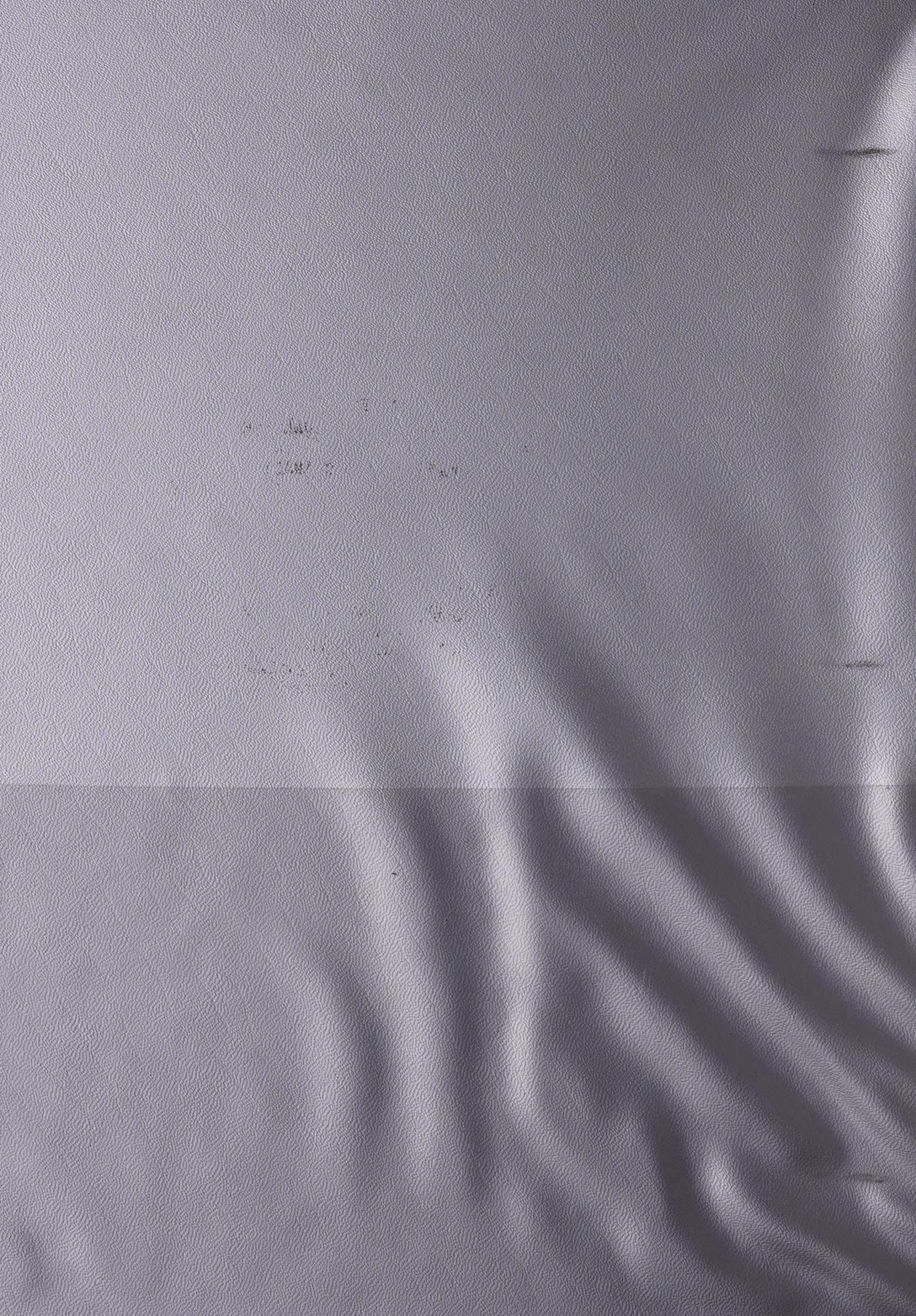
ARIUM
CORPORATION



ARIUM CORPORATION

OPERATING MANUAL FOR

ML4100 LOGIC ANALYZER

June 2, 1986, edition

(Revises and replaces
May 10, 1985, edition)

Copyright 1984, 1985, & 1986

Arium Corporation · 1931 Wright Circle, Anaheim, CA 92806-6052 · Tel.: (714) 978-9531 · Telex: 754903 ARIUM UD

REVISION RECORD

The following pages have been revised since the publication of this manual (June 2, 1986). On the revised pages, each revised line is marked with a vertical bar (|) in the right margin.

III. Display and Setup of Operating Parameters

4 8-25-88

IX. Setup Memory Option and Setup/Data Memory Option

13 8-25-88

REVISION RECORD

The following pages have been revised since the publication of this manual (June 2, 1986). On the revised pages, each revised line is marked with a vertical bar (|) in the right margin.

III. Display and Setup of Operating Parameters

4 8-25-88

IX. Setup Memory Option and Setup/Data Memory Option

13 8-25-88

TABLE OF CONTENTS
(followed by LIST OF FIGURES)

I. INTRODUCTION

A. ML4100 DESCRIPTION I- 3

B. OVERALL SYSTEM FUNCTION I- 4
1. RECORDS DIGITAL DATA I- 4
2. DISPLAYS DATA IN THREE MODES I- 4
3. STANDARD ACCESSORY: 32-CHANNEL LOGIC POD (LP-320) I- 4
4. OPTIONAL ACCESSORIES (available for separate purchase) I- 4
a. 4-Channel (Glitch-Capture) Logic Pod (LP-040) I- 4
b. STD Bus Pod (8P-080) I- 4
c. Microprocessor Pods I- 5
d. ROM Emulator Module (RE-016) I- 5
e. RS-232C Communication Control Card (RS-232) I- 5
f. Setup/Data Memory Option (AB-032) I- 5
g. Setup Memory Option (SM-080) I- 5

C. SYSTEM SPECIFICATIONS (See Appendix A) I- 5

Figure 1: FRONT PANEL I- 6

D. FRONT-PANEL FUNCTIONS I- 7
1. CONNECTOR RECEPTACLE I- 7
2. KEYBOARD AND CRT SCREEN I- 7

Figure 2: BACK PANEL I- 8

E. BACK-PANEL FUNCTIONS I- 9
1. POWER SWITCH (ON/OFF) I- 9
2. P1/DATA INTERFACE I- 9
3. P2/ROM EMULATOR I- 9
4. BRIGHTNESS ADJUSTING SCREW I- 9

F. AUDIBLE WARNINGS I- 9
1. ONE BEEP I- 9
2. TWO BEEPS I- 9
3. THREE BEEPS I- 9
4. CONTINUOUS BEEPS I- 9

II. CONNECTION TO SYSTEM UNDER TEST

Figure 3: COLOR CODES OF PROBES FOR 32-CHANNEL LOGIC POD II- 2

A. CONNECTION VIA LOGIC POD II- 3
1. 32-CHANNEL LOGIC POD PROBES II- 4
2. 4-CHANNEL LOGIC POD PROBES II- 4
3. CONNECTING LOGIC POD TO SYSTEM UNDER TEST II- 4

B. CONNECTION VIA MICROPROCESSOR PODS II- 5

C. CONNECTION VIA STD BUS POD II- 5

D. ROM EMULATOR MODULE CONNECTION II- 5

III. DISPLAY AND SETUP OF OPERATING PARAMETERS

Figure 1: FRONT PANEL (repeated) III- 2

A. INTRODUCTION III- 3

B. SYSTEM STATUS SETUP SCREEN (VIA "STATUS" KEY) III- 4
Figure 4: STATUS SETUP SCREEN III- 5

1. CLOCK III- 6
2. CONFIGURATION III- 6
3. (TRIGGER) DELAY III- 6
4. (TRIGGER) SEQUENCE III- 6
5. TRIGGER III- 6
6. FORMAT III -7

C. CLOCK SETUP SCREEN (VIA "CLOCK" KEY) III- 8
Figure 5: CLOCK SETUP SCREEN III- 8

1. SELECTING CLOCK SOURCE III- 8
2. USING AN INTERNAL CLOCK III- 8
3. USING AN EXTERNAL CLOCK III- 9
   a. Clock Edge III- 9
   b. Clock Qualifier III- 9

D. CONFIGURATION SETUP SCREEN (VIA "CONFIG" KEY) III-10
Figure 6: CONFIGURATION SETUP SCREEN III-10

1. RECORD WIDTH III-10
2. DATA QUALIFICATION (in the 32-channel/bit mode only) III-11
   a. Combinational Data Qualification III-11
   b. State Data Qualification III-11

E. TRIGGER SETUP SCREEN (VIA "TRIG" KEY) III-12
Figure 7: TRIGGER SETUP SCREEN III-12

F. TRIGGER SEQUENCE SETUP SCREEN (VIA "SEQ" KEY) III-14
Figure 8: TRIGGER SEQUENCE SETUP SCREEN III-14

1. PREDEFINED TRIGGER SEQUENCES III-15
2. USER-DEFINED TRIGGER SEQUENCE III-16
   Figure 9: USER-DESIGNED SEQUENCE SETUP SCREEN III-16

G. TRIGGER DELAY SETUP SCREEN (VIA "DELAY") KEY III-18
Figure 10: TRIGGER DELAY SETUP SCREEN III-19

H. DISPLAY FORMAT SETUP SCREEN (VIA "FORMAT" KEY) III-20
Figure 11: DISPLAY FORMAT SETUP SCREEN III-20

1. FORMAT FOR TIMING DISPLAY MODE III-20
2. FORMAT FOR STATE DISPLAY MODE III-21
3. DISPLAY FORMAT FOR DISASSEMBLY (VIA MICROPROCESSOR PODS) AND OTHER POD MODES III-21
   Figure 12: POD MODE DESIGNATIONS III-22

I. SPECIAL FUNCTIONS SETUP SCREEN (VIA "S. FUNC" KEY) III-23
Figure 13: SPECIAL FUNCTIONS III-23

1. MAINFRAME SELF-TEST (STANDARD) III-24
2. STANDARD PROBE SET COLOR CODES III-24
3. RS-232C PORT CONFIGURATION III-24
4. ROM EMULATOR TRANSFERS III-24
5. ROM EMULATOR CONFIGURATION III-24
6. ROM EMULATOR EDIT III-24
7. SETUP STORAGE III-24
8. B MEMORY STORAGE III-24
9. COMPARE III-24
10. SEARCH III-24
11. EDIT B MEMORY III-24

IV. OPERATION

A. GENERAL IV- 2
1. RECORDING DATA IV- 2
2. DISPLAYING RECORDED DATA IV- 2
3. PRINTING DISPLAYED DATA IV- 2
4. MOVING THE CURSOR AND SCROLLING IV- 3

B. TIMING DISPLAY MODE IV- 4
Figure 14: TIMING DISPLAY SCREEN IV- 5

C. STATE DISPLAY MODE IV- 6
Figure 15: STATE DISPLAY SCREEN IV- 6

D. DISASSEMBLY DISPLAY MODE (OPTIONAL) IV- 7
Figure 16: DISASSEMBLY DISPLAY SCREEN IV- 7

V. FOUR-CHANNEL LOGIC POD (LP-040)

A. SPECIFICATIONS V- 2

B. OPERATION V- 2
1. PROBES V- 2
2. MEASURING THRESHOLD VOLTAGE V- 3
3. SETUP V- 3

VI. 8-BIT MICROPROCESSOR PODS

A. GENERAL VI- 2

B. 8I-080 MICROPROCESSOR POD (for Intel 8085, 8035/8039/8040, and 8031/8032) VI- 3

C. 8M-080 MICROPROCESSOR POD (for Motorola 6800, 6802, and 6808) VI- 4

D. 8M-089 MICROPROCESSOR POD (for Motorola 6809 and 6809E) VI- 5

E. 8R-065 MICROPROCESSOR POD (for Rockwell 6502, 6512, 65C02, 65C102, and 65C112) VI- 6

F. 8Z-080 MICROPROCESSOR POD (for Zilog Z80) VI- 7

G. 8N-080 MICROPROCESSOR POD (for National Semiconductor NSC800) VI- 8

VII. STD BUS POD (8P-080)

A. INTRODUCTION VII- 2

B. POD MODES VII- 2

C. JUMPERING OPTIONS VII- 3
1. DATA LINES 24-31 VII- 3
2. REFRESH CYCLES VII- 3
3. INTRQ* AND BUSRQ* VII- 4

D. ASYNCHRONOUS OPERATION VII- 4

VIII. ROM EMULATOR MODULE (RE-016)

A. INTRODUCTION VIII- 3

B. SETUP AND EDITING VIII- 3
1. GENERAL VIII- 3
2. CONFIGURATION OF RS-232C SERIAL PORT VIII- 4
Figure 17: RS-232C SERIAL PORT CONFIGURATION VIII- 5
3. DATA TRANSFERS TO AND FROM ROM EMULATOR VIII- 6
Figure 18: ROM EMULATOR TRANSFERS VIII- 7
4. ROM EMULATOR CONFIGURATION VIII- 8
Figure 19: ROM EMULATOR CONFIGURATION VIII- 9
5. EDITING DATA IN THE ROM VIII-10
Figure 20: ROM EMULATOR EDIT VIII-10

C. HARDWARE CONSIDERATIONS VIII-11

D. ERROR CODES VIII-12

IX. SETUP MEMORY OPTION (SM-080) AND SETUP/DATA MEMORY OPTION (AB-032)

A. INTRODUCTION IX- 3
1. SETUP MEMORY OPTION (SM-080) IX- 3
2. SETUP/DATA MEMORY OPTION (AB-032) IX- 3
3. INSTALLATION IX- 3

B. OVERALL OPERATION IX- 4

1. SETUP STORAGE IX- 4
a. Saving Setups IX- 5
Fig. 21a: SETUP STORAGE SCREEN (Cursor in Action) IX- 5
b. File Names IX- 6
Fig. 21b: SETUP STORAGE SCREEN (Cursor in Name) IX- 6
c. Status (Protected, Unprotected, Available) IX- 7
Fig. 21c: SETUP STORAGE SCREEN (Cursor in Status) IX- 7

2. B MEMORY STORAGE IX- 8
a. B Memory Configuration IX- 8
b. Copying Data IX- 8
c. Storing Data in Saved B Memory IX- 9
Fig. 22a: B MEMORY STORAGE SCREEN (Cursor in Action) IX- 9
d. B Memory File Names IX-10
Fig. 22b: B MEMORY STORAGE SCREEN (Cursor in Name) IX-10
e. Status (Protected, Unprotected, Available) IX-11
Fig. 21c: B MEMORY STORAGE SCREEN (Cursor in Status) IX-11

3. COMPARING DATA (AB-032 only) IX-12
a. Single Compare IX-12
Fig. 23a: COMPARE SCREEN IX-13
Fig. 23b: COMPARE SCREEN (Single Compare Results) IX-14
b. Continuous Compare IX-14
Fig. 23c: COMPARE SCREEN (Contin. Compare Results) IX-15
c. Autosequencing IX-16
Fig. 23d: AUTOSEQUENCING RESULTS IX-17

4. SEARCH (AB-032 only) IX-18
Figure 24a: SEARCH SCREEN IX-18
Figure 24b: SEARCH SCREEN (Results) IX-19

5. B MEMORY EDIT (AB-032 only) IX-20
Figure 25: B MEMORY EDIT SCREEN IX-20

6. B MEMORY DISPLAY IX-21

C. EXECUTION TIMES IX-22

1. SETUP STORAGE TIMES IX-22
2. B MEMORY STORAGE TIMES IX-22
3. COPY A TO B EXECUTION TIMES IX-22
4. SEARCH EXECUTION TIMES IX-23
5. COMPARE EXECUTION TIMES IX-23

X. 16M-680 MICROPROCESSOR POD

A. GENERAL X- 2
1. SPECIAL REQUIREMENTS FOR ML4100 TO ACCOMMODATE 16-BIT MICROPROCESSOR PODS X- 2
2. OVERVIEW OF 16-BIT MICROPROCESSOR POD OPERATION X- 2

B. HARDWARE X- 3
1. SWITCH SETTINGS X- 3
2. SIGNAL ASSIGNMENTS X- 4
Fig. 26: SIGNAL ASSIGNMENTS--OVERVIEW X- 4
Fig. 27: DISPLAY FORMAT SCREEN X- 4
Fig. 28: SIGNAL ASSIGNMENTS--DETAILED X- 5
Fig. 29: DEFAULT NUMBER BASES FOR DISPLAYS X- 6
Fig. 30: QUICK REFERENCE LIST FOR SOME STATUS & BINARY SIGNALS X- 6
Fig. 31: DETAILED LIST OF BINARY AND STATUS SIGNALS X- 7
3. STATUS CODES X- 8
Fig. 32: TRIGGER SCREEN (WITH 16M-680 POD) X- 8
Fig. 33: STATUS CODE EXAMPLES X- 9

C. TRANSPARENT MODE X- 9

D. FILTERED MODE X-10
1. GENERAL X-10
2. HARDWARE CONSIDERATIONS X-10
a. General X-10
b. FIFO Memory X-10
c. Unremoved Prefetches X-11
d. Self-Synchronization X-11

E. JUMP MODE X-12

F. TRIGGERING CONSIDERATIONS X-12
1. TRIGGERING SEQUENCE LIMITATIONS IN 62-BIT OPERATION X-12
2. DATA QUALIFICATION X-13
3. UNDESIRED TRIGGERING ON PREFETCHES IN TRANSPARENT MODE X-13

G. DATA DISPLAY X-14
1. GENERAL X-14
Fig. 34: STATE DISPLAY (ALL MODES) X-14
Fig. 35: DISASSEMBLY DISPLAY (TRANSPARENT AND FILTERED MODES) X-15
Fig. 36: DISASSEMBLY DISPLAY (JUMP MODE) X-15
2. KEY FUNCTIONS ON THE DISASSEMBLY SCREEN X-16
a. f1 Key (Position Cursor) X-16
b. Up/Down Arrow Keys (Scroll through Buffer) X-16
c. f2 Key (Scroll Speed) X-16
d. f3 Key (Data Cycles On/Off) X-16
e. f4 Key (Print Screen) X-16
f. DISASM Key (Display Trigger Condition) X-16
3. DISPLAY FORMATS X-17
a. Instruction Display X-17
b. Data Cycles Display X-17
c. Specific Instruction Display Techniques X-17

H. DISASSEMBLY SOFTWARE X-19

XI. 16I-086 MICROPROCESSOR POD

A. GENERAL XI- 2
1. SPECIAL REQUIREMENTS FOR THE ML4100 TO ACCOMMODATE THE POD XI- 2
2. OVERVIEW OF 16-BIT MICROPROCESSOR POD OPERATION XI- 3

B. HARDWARE XI- 4
1. SWITCH SETTINGS XI- 4
2. SIGNAL ASSIGNMENTS XI- 5
Fig. 37: DESIGNATIONS OF 16I-086 PROCESSOR/POD MODES XI- 5
Fig. 38: SIGNAL CLASS ASSIGNMENTS--OVERVIEW XI- 5
Fig. 39: STATUS SIGNAL ASSIGNMENTS XI- 6
Fig. 40: STATUS SIGNALS WHOSE NAMES DO NOT CORRESPOND TO 8086/8088 SIGNALS XI- 7
Fig. 41: BINARY FIELD SIGNAL ASSIGNMENTS XI- 7
3. STATUS CODES XI- 8
Fig. 42: TYPICAL STATUS CODES FOR TRIGGERING ON SPECIFIC CYCLE TYPES XI- 8
Fig. 43: TYPICAL STATUS CODES FOR TRIGGERING ON SPECIFIC MEMORY SEGMENT ACCESSES XI- 8
4. POD CONNECTION ERROR WARNING XI- 9

C. MINIMUM PROCESSOR MODE (TRANSPARENT POD MODE ONLY) XI-10

D. MAXIMUM PROCESSOR MODE WITH TRANSPARENT POD MODE XI-11

E. MAXIMUM PROCESSOR MODE WITH FILTERED POD MODE XI-11
1. 8086F AND 8088F MODES XI-11
2. 8086F MODE; REPACKING INSTRUCTION BYTES XI-11

F. MAXIMUM PROCESSOR MODE WITH JUMP POD MODE XI-12

G. TRIGGERING AND DATA QUALIFICATION XI-13
1. TRIGGERING SEQUENCE LIMITATIONS IN 62-BIT OPERATION XI-13
2. DATA QUALIFICATION IN 62-BIT OPERATION XI-14
3. TRIGGERING METHODS WITH PREFETCHING OR BYTE REPACKING XI-14
a. Overfetch Problems in the Transparent Mode XI-14
b. Triggering Problems Due to Instruction Byte Separation within a Word XI-15

H. DISPLAY XI-16
1. GENERAL XI-16
Fig. 44: DISASSEMBLY DISPLAY (TRANS. & FILT. MODES) XI-17
Fig. 45: DISASSEMBLY DISPLAY (JUMP MODE) XI-17
2. KEY FUNCTIONS ON THE DISASSEMBLY DISPLAY SCREEN XI-18
a. f1 Key (Position Cursor) XI-18
b. Up/Down Arrow Keys (Scroll through Buffer) XI-18
c. f2 Key (Scroll Speed) XI-18
d. f3 Key (Data Cycles On/Off) XI-18
e. f4 Key (Print Screen) XI-18
f. DISASM Key (Display Trigger Condition) XI-18
g. X Key XI-18
3. DISASSEMBLY DISPLAY FORMATS XI-19
a. Instruction Display XI-19
b. Data Cycles Display XI-19
c. Interrupts XI-20
d. Question Marks XI-20
e. "UNABLE TO DISASSEMBLE" Messages XI-20

XII. 8RS-232 SERIAL DATA ANALYSIS POD

A. INTRODUCTION XII- 3
1. OVERVIEW XII- 3
2. CONNECTION XII- 3
Fig. 46: 8RS-232 POD CONNECTION

B. TRIGGERING XII- 4
Fig. 47: TRIGGER SETUP SCREEN XII- 4
1. TRIGGER SETUP SCREEN XII- 4
a. ERR (Errors) XII- 4
b. TAG (RS-232C Protocol Signals) XII- 5
c. DAT (Data) XII- 5
d. STS (Status) XII- 5
e. R/T (Receive/Transmit Pin Selection) XII- 5
f. BIN (Binary) XII- 5
2. SELECTION OF TRIGGER EVENT PARAMATERS XII- 5

C. DISPLAY FORMATS XII- 6
1. GENERAL XII- 6
Fig. 48: DISPLAY FORMAT SETUP SCREEN XII- 6
2. TIMING DISPLAY SCREEN XII- 7
Fig. 49: TIMING DISPLAY SCREEN XII- 7
3. STATE DISPLAY SCREEN XII- 8
Fig. 50: STATE DISPLAY SCREEN XII- 8
a. ERR XII- 8
b. TAG XII- 8
c. STS XII- 8
d. BIN XII- 8
4. DISASSEMBLY DISPLAY SCREEN XII- 9
Fig. 51: DISASSEMBLY DISPLAY SCREEN XII- 9
a. RS-232 DISPLAY SCREEN XII- 9
b. DIR XII- 9
c. HEX and ASC XII- 9
d. TAG XII- 9
e. ERR XII- 9
f. SYNC XII- 9
5. TEXT DISPLAY SCREEN XII-10
Fig. 52: TEXT DISPLAY SCREEN XII-10

D. SPECIAL FUNCTIONS XII-11
Fig. 53: HEX-TO-ASCII CONVERSION TABLE XII-11

E. RS-232C COMMON PIN ASSIGNMENTS XII-12

APPENDIX A: SPECIFICATIONS

A. DATA INPUTS A- 2
B. CLOCK (Single) A- 2
C. TRIGGERING A- 3
D. DATA QUALIFICATION (32-channel mode only) A- 3
E. MEMORY A- 3
F. STATE DISPLAY MODE A- 3
G. TIMING DISPLAY MODE A- 4
H. DISASSEMBLY DISPLAY MODE A- 4
I. SCROLLING A- 4
J. CRT A- 4
K. PHYSICAL MEASUREMENTS A- 4
L. INPUT POWER A- 4
M. ENVIRONMENTAL A- 5
N. ROM EMULATOR MODULE (optional) (RE-016) A- 5
O. 32-CHANNEL LOGIC POD (LP-320) A- 5
P. OPTIONAL PODS A- 5

APPENDIX B: ASCII
(USA STANDARD CODE FOR INFORMATION EXCHANGE)

APPENDIX C: INTEL HEX FORMAT

APPENDIX E: ERROR MESSAGES

INDEX

LIST OF FIGURES

(INTRODUCTION)
1: FRONT PANEL I- 6 & III- 2
2: BACK PANEL I- 8

(CONNECTION TO SYSTEM UNDER TEST)
3: COLOR CODES OF PROBES FOR 32-CHANNEL LOGIC POD II- 2

(DISPLAY AND SETUP OF OPERATING PARAMETERS)
4: (SYSTEM) STATUS SETUP SCREEN III- 5
5: CLOCK SETUP SCREEN (VIA "CLOCK" KEY) III- 8
6: CONFIGURATION SETUP SCREEN (VIA "CONFIG" KEY) III-10
7: TRIGGER SETUP SCREEN (VIA "TRIG" KEY) III-12
8: TRIGGER SEQUENCE SETUP SCREEN (VIA "SEQ" KEY) III-14
9: USER-DEFINED SEQUENCE SETUP SCREEN III-16
10: TRIGGER DELAY SETUP SCREEN (VIA "DELAY" KEY) III-19
11: DISPLAY FORMAT SETUP SCREEN (VIA "FORMAT" KEY) III-20
12: POD MODE DESIGNATIONS III-22
13: SPECIAL FUNCTIONS III-23
14: TIMING DISPLAY SCREEN IV- 5
15: STATE DISPLAY SCREEN IV- 6
16: DISASSEMBLY DISPLAY SCREEN IV- 7

(ROM EMULATOR MODULE)
17: RS-232C SERIAL PORT CONFIGURATION VIII- 5
18: ROM EMULATOR TRANSFERS VIII- 7
19: ROM EMULATOR CONFIGURATION VIII- 9
20: ROM EMULATOR EDIT VIII-10

(SETUP MEMORY OPTION AND SETUP/DATA MEMORY OPTION)
21a: SETUP STORAGE SCREEN (Cursor in Action Field) IX- 5
21b: SETUP STORAGE SCREEN (Cursor in Name Field) IX- 6
21c: SETUP STORAGE SCREEN (Cursor in Status Field) IX- 7
22a: B MEMORY STORAGE SCREEN (Cursor in Action Field) IX- 9
22c: B MEMORY STORAGE SCREEN (Cursor In Name Field) IX-10
22d: B MEMORY STORAGE SCREEN (Cursor In Status Field) IX-11
23a: COMPARE SCREEN IX-13
23b: COMPARE SCREEN (Single Compare Results) IX-14
23c: COMPARE SCREEN (Continuous Compare Results) IX-15
23d: AUTOSEQUENCING RESULTS IX-17
24a: SEARCH SCREEN IX-18
24b: SEARCH SCREEN (Results) IX-19
25: B MEMORY EDIT SCREEN IX-20

(16M-680 MICROPROCESSOR POD)
26: SIGNAL ASSIGNMENTS--OVERVIEW X- 2
27: DISPLAY FORMAT SCREEN X- 4
28: SIGNAL ASSIGNMENTS--DETAILED X- 5
29: DEFAULT NUMBER BASES FOR DISPLAYS X- 6
30: QUICK REFERENCE LIST FOR SOME STATUS & BINARY SIGNALS X- 6
31: DETAILED LIST OF BINARY AND STATUS SIGNALS X- 7
32: TRIGGER SCREEN (WITH 16M-680 POD) X- 8
33: STATUS CODE EXAMPLES X- 9
34: STATE DISPLAY (ALL MODES) X-14
35: DISASSEMBLY DISPLAY (TRANSPARENT AND FILTERED MODES) X-15
36: DISASSEMBLY DISPLAY (JUMP MODE) X-15

(16I-086 MICROPROCESSOR POD)
37: DESIGNATIONS OF 16I-086 PROCESSOR/POD MODES XI- 5
38: SIGNAL CLASS ASSIGNMENTS--OVERVIEW XI- 5
39: STATUS SIGNAL ASSIGNMENTS XI- 6
40: STATUS SIGNALS WHOSE NAMES DO NOT CORRESPOND TO 8086/8088 SIGNALS XI- 7
41: BINARY FIELD SIGNAL ASSIGNMENTS XI- 7
42: TYPICAL STATUS CODES FOR TRIGGERING ON SPECIFIC CYCLE TYPES XI- 8
43: TYPICAL STATUS CODES FOR TRIGGERING ON SPECIFIC MEM. SEG. ACCESSES XI- 8
44: DISASSEMBLY DISPLAY (TRANSPARENT AND FILTERED MODES) XI-17
45: DISASSEMBLY DISPLAY (JUMP MODE) XI-17

(8RS-232 SERIAL DATA ANALYSIS POD)
46: 8RS-232 POD CONNECTION XII- 3
47: TRIGGER SETUP SCREEN XII- 4
48: DISPLAY FORMAT SETUP SCREEN XII- 6
49: TIMING DISPLAY SCREEN XII- 7
50: STATE DISPLAY SCREEN XII- 8
51: DISASSEMBLY DISPLAY SCREEN XII- 9
52: TEXT DISPLAY SCREEN XII-10
53: HEX-TO-ASCII CONVERSION TABLE XII-11

I. INTRODUCTION

(blank)

A. ML4100 DESCRIPTION

The Arium ML4100 Logic Analyzer provides full-featured logic analysis for the debugging and troubleshooting of digital and microprocessor-based products, including complex triggering with state, timing, and (optionally) microprocessor disassembly displays. Its basic function is to capture (record) the digital data on several signal lines, repetitively, and to do this synchronously or asynchronously with the clock of the system under test. High-speed, asynchronous operation is a powerful aid when solving hardware logic timing problems, while low-speed, synchronous operation is useful for debugging software.

The ML4100 operates at speeds of up to 100 MHz, at data widths of up to 32 channels, and at a memory depth of up to 8000 samples per channel. (At 100 MHz, the maximum number of channels is 4; when using 32 channels, the maximum speed is 12.5 MHz; and the maximum memory depth of 8000 samples per channel is available only when using up to 4 channels.)

Data and clock qualifiers can be used, as well as a trigger delay of up to 50,000 clocks. (Data qualification is available only with 32-channel operation.) The optional ROM Emulator Module (RE-016) provides in-circuit emulation and editing of code stored in 2716, 2732, 2764, and 27128 PROMs; it includes an RS-232C interface, which permits downloading or uploading of code. The expanded memory option (Setup/Data Memory, AB-032) permits both machine setups and data to be uploaded or downloaded, as well as stored while power is off.

The ML4100 contains a 5-inch, green-screen CRT which displays up to twelve channels of timing diagrams (with labelling) and has cursors for measuring time periods. The state display can be formatted and reformatted to achieve almost any conceivable digit combination in several number bases: ASCII, binary, octal, decimal, or hexadecimal.

Other options are microprocessor pods with one-clip connection to most popular microprocessors; the resulting disassembly display shows mnemonics for the various instruction and data sequences.

Four trigger words are available, and can be defined in binary, octal, decimal, or hexadecimal number base. These four words can be combined to form the trigger condition in many different ways. Nineteen trigger sequences are predefined and selectable, and the user may define his own sequence as well. All predefined sequences are available in the 32-channel mode, but these sequences are somewhat more limited in the other modes.

The ML4100 has been designed for simple setup and ease of operation. All setup and display screens contain complete prompting information which is self-explanatory for most situations. This enables the user to make productive use of this logic analyzer with a minimum of experience and, we hope, only infrequent reference to this manual.

B. OVERALL SYSTEM FUNCTIONS

1. RECORDS DIGITAL DATA AT:

* Speeds (clock/sampling rates) of up to 100 MHZ (10 nsec between samples) (4 channels maximum at 100MHz)

* Data (word) widths of 4 to 32 channels (bits) (12.5 MHz maximum at 32 channels)

* Record depth (memory) of up to 8000 samples per channel (4 channels maximum at 8000 samples)

2. DISPLAYS DATA IN THREE MODES:

* Timing (as on an oscilloscope)

* State (a formatted word list)

* Disassembly (microprocessor assembly code instructions)--optional

3. STANDARD ACCESSORY: 32-CHANNEL LOGIC POD (LP-320)

The ML4100 includes a 32-Channel Logic Pod (with Standard Probe Set) at no extra charge. This pod supports analyzer operation at all speeds and recording widths, with internal or external clocking; clock qualification is available with external clock. (See also Section II.A, re connection of this pod to the ML4100.)

4. OPTIONAL ACCESSORIES (available for separate purchase)

a. 4-Channel (Glitch-Capture) Logic Pod (LP-040)

This pod operates at any speed of up to 100 MHz and a data width of 4 channels to provide advanced timing analysis with high-speed clock qualification and glitch capture. (See also Section II.A, re connection.)

b. STD Bus Pod (8P-080)

This circuit board provides connection to an STD bus structure. It plugs into the bus as an extender card (not as a replacement card), and can thus be used in a backplane which is fully occupied. (See Section VII for further details.)

c. Microprocessor Pods

Microprocessor pods provide one-clip connection between the ML4100 and most popular microprocessors. The following such pods are available:

Catalog No.:	Microprocessors Supported:
8I-080	Intel 8085, 8031/8032, & 8035/8039/8040
8M-080	Motorola 6800/6802/6808
8M-089	Motorola 6809/6809E
8N-080	National Semiconductor NSC800
8R-065	Rockwell 6502, 6512 65C02/65C102, & 65C112
8Z-080	Zilog Z80, Z80A, Z80B, & Z80C
16M-680	Motorola 68000/68010

(See Section VI for further details.)

d. ROM Emulator Module (RE-016)

The ROM Emulator module emulates one to four Read-Only Memories per pod, up to a total of 16 Kbytes of memory, and is configurable for 2716, 2732, 2764, or 27128 PROMs. One or two pods may be used for a total of 32 Kbytes of emulation memory. The simulated ROM memory can be uploaded or downloaded, in several popular data transfer formats, to or from other equipment (such as computers and PROM programmers) via an RS-232C port on the back panel. Memory is then displayed on the ML4100 screen and can be changed directly with front-panel key entries.

(See Section VIII for further details.)

e. RS-232C Communication Control Card (RS-232)

This circuit board is contained within the ROM Emulator Module, but may also be purchased separately when the user wishes an interface between the ML4100 and other equipment. such as a computer or a printer. With this board, the user may specify operating parameters (baud rate, format, handshakes, etc.) from the ML4100 front panel.

f. Setup/Data Memory Option (AB-032)

This option provides the ML4100 with nonvolatile memory so that machine setups and capture data are permanently stored (even while the power is off) via an internally installed, electrically eraseable ROM board.

g. Setup Memory Option (SM-080)

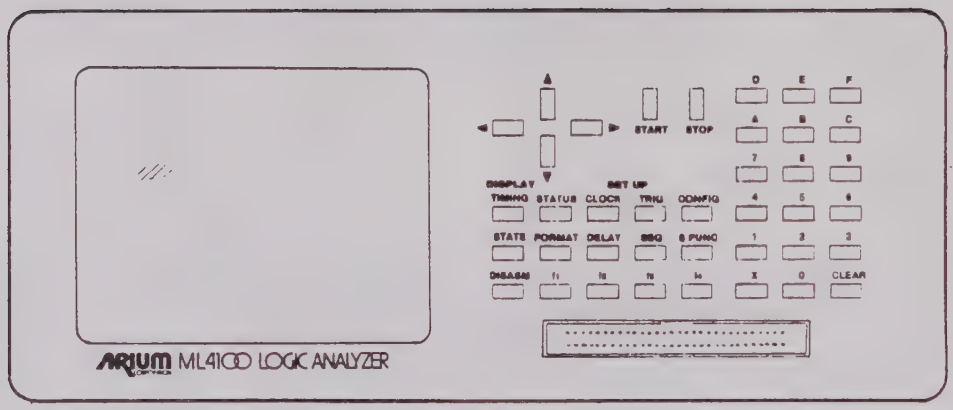
This option provides the ML4100 with nonvolatile memory of machine setups only. (See Item f above.)

C. SYSTEM SPECIFICATIONS

Detailed system specifications are given in Appendix A.

Figure 1

FRONT PANEL



D. FRONT-PANEL FUNCTIONS (See Figure 1)

1. **CONNECTOR RECEPTACLE** (At botton of right side of panel) -- For connecting a selected pod to the ML4100.

WARNING: DO NOT PLUG OR UNPLUG THE POD CONNECTOR WHILE UNIT IS POWERED UP.

(See Section II.A for details of use.)

2. KEYBOARD AND CRT SCREEN

All operating functions are controlled from the front-panel keyboard. All keys will repeat if held down; use a quick and sure touch to avoid unwanted repeating.

a. **START and STOP Keys** -- To start and stop recording of data. (To turn on the ML4100, use the POWER Switch (ON/OFF) on the back panel.)

b. **Hexadecimal Keypad** (on the right) -- To set and change recording and display parameters.

c. **Four Arrow Keys** -- To move the cursor on setup screens, and to scroll through the data on display screens.

d. Three "DISPLAY" Keys (TIMING, STATE, and DISASM) -- To select the data display mode.

e. **Twelve "SETUP" Keys** -- To access setup screens for specifying parameters for data recording and data display formatting.

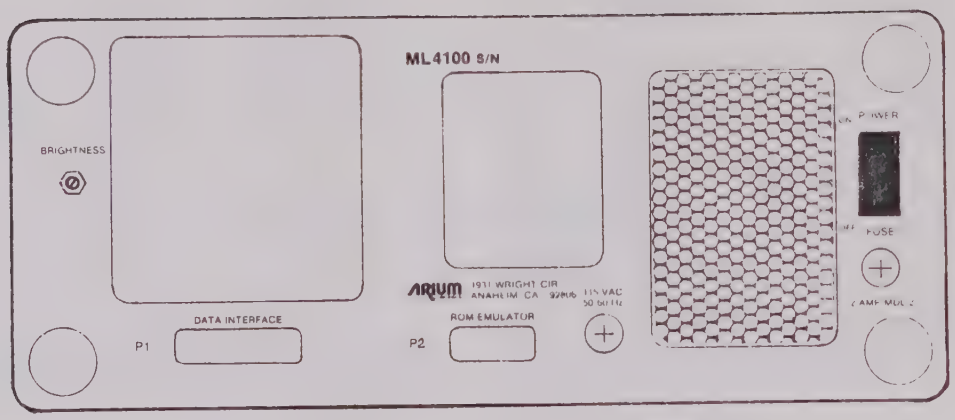
f. CRT Screen -- To display data and to set up recording and display parameters. Any setup screen may be accessed with only a single keystroke. Changeable fields are reverse-videoed and the cursor position blinks; the cursor automatically skips over nonchangeable fields. (The CRT has a 5-inch green screen.)

To change any setup or operating parameter, simply position the cursor into its field, then enter the desired parameter via the keypad; the new parameter (and any other value determined from it) will be updated and displayed. (Actual update occurs when the screen or field is exited via the setup keys or cursor.)

(See also Section IV.A.3, Moving the Cursor and Scrolling.)

Figure 2

BACK PANEL



E. BACK-PANEL FUNCTIONS (see Figure 2)

1. POWER SWITCH (ON/OFF)

2. P1/DATA INTERFACE -- Connector receptacle for RS-232C interface with a host computer or other equipment.

(See Section VIII, ROM Emulator Module, for details about use of this interface.)

3. P2/ROM EMULATOR -- Connector receptacle for ROM Emulator Module.

(See Section VIII for details about this module.)

4. BRIGHTNESS ADJUSTING SCREW -- To vary the intensity of the CRT screen.

F. AUDIBLE WARNINGS

1. ONE BEEP -- Warns that a wrong key has been depressed (including trying to move the cursor past the limits of the field or screen), and has been ignored by the system. For example, depressing a numeric key on a display screen will usually produce one beep.

2. TWO BEEPS -- Signal that data has stopped being recorded, i.e., that the trigger condition has occurred and the trigger delay has been satisfied.

3. THREE BEEPS -- Warn that setup entries are inconsistent, or that there are pod status errors. The screen will display an explanatory error message, such as "ALREADY STARTED" if the START key is depressed a second time, "POD CONNECTION ERROR" if the connection to the unit under test is wrong, "NO DATA RECORDED" if the STOP key is depressed before any data has been recorded, etc.

4. CONTINUOUS BEEPS -- Indicate that a wrong key is being depressed continuously and is repeating, or, at powerup, that a key is stuck.

(Blank)

II. CONNECTION TO SYSTEM UNDER TEST

Figure 3
COLOR CODES OF PROBES FOR 32-CHANNEL LOGIC POD

	BODY: (tip)	PLUNGER: (cap)	WIRE:	CONNECTOR/PIN:
External Clock	Red	Red	Red*	P1B-16
Clock Qualifier	Blue	Blue	Blue*	P1B-14
Grounds (8)	Black	Black	Black*	
Probe 0	Black	Grey	Black	P1B-2
Probe 1	Black	Grey	Brown	P1B-3
Probe 2	Black	Grey	Red	P1B-5
Probe 3	Black	Grey	Orange	P1B-6
Probe 4	Black	Grey	Yellow	P1B-8
Probe 5	Black	Grey	Green	P1B-9
Probe 6	Black	Grey	Blue	P1B-11
Probe 7	Black	Grey	Violet	P1B-12
Probe 8	Black	Grey	Grey	P1A-2
Probe 9	Black	Grey	White	P1A-3
Probe 10	Brown	Grey	Black	P1A-4
Probe 11	Brown	Grey	Brown	P1A-5
Probe 12	Brown	Grey	Red	P1A-6
Probe 13	Brown	Grey	Orange	P1A-7
Probe 14	Brown	Grey	Yellow	P1A-8
Probe 15	Brown	Grey	Green	P1A-9
Probe 16	Brown	Grey	Blue	P1A-10
Probe 17	Brown	Grey	Violet	P1A-11
Probe 18	Brown	Grey	Grey	P1A-12
Probe 19	Brown	Grey	White	P1A-13
Probe 20	Red	Grey	Black	P1A-15
Probe 21	Red	Grey	Brown	P1A-16
Probe 22	Red	Grey	Red	P1A-17
Probe 23	Red	Grey	Orange	P1A-18
Probe 24	Red	Grey	Yellow	P1A-19
Probe 25	Red	Grey	Green	P1A-20
Probe 26	Red	Grey	Blue	P1A-21
Probe 27	Red	Grey	Violet	P1A-22
Probe 28	Red	Grey	Grey	P1A-23
Probe 29	Red	Grey	White	P1A-24
Probe 30	Orange	Grey	Black	P1A-25
Probe 31	Orange	Grey	Brown	P1A-26

* Optional disconnectible probes have yellow sleeves at end, to distinguish these wires from the data probes when the probe tips have been removed.

A. CONNECTION VIA LOGIC POD

WARNING: DO NOT PLUG THE POD CONNECTOR INTO, OR UNPLUG IT FROM, THE FRONT-PANEL CONNECTOR RECEPTACLE WHILE THE ML4100 IS POWERED UP.

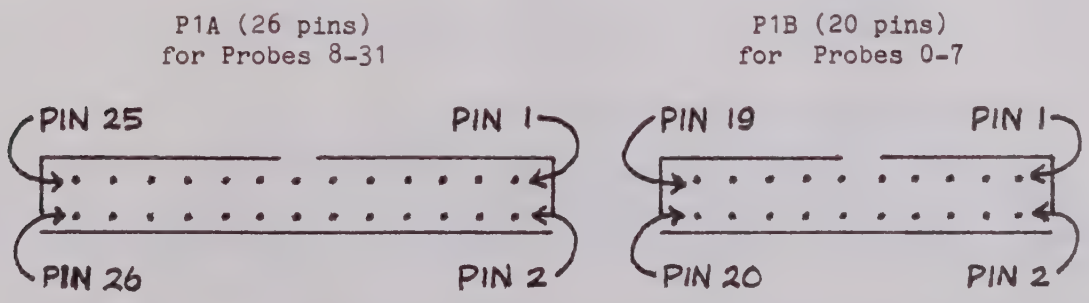
NOTE: Connection procedures are the same for both the 32-Channel (Standard) Logic Pod (LP-320) and the 4-Channel (Glitch-Capture) Logic Pod (LP-040).

The 32-channel pod has 42 wires (16 on one probe set and 26 on the other probe set), connectored and color-coded as described below. The 4-channel pod has 6 twisted wire pairs terminated in E-Z Hook test clips. Each signal lead has a resistor in the lead and adjacent to the signal clip to reduce reflection.

Each probe has 3 parts: the tip (body), the cap (plunger), and the wire; each part may be one of ten colors. Clock Probes have all three parts colored red; Clock Qualifier Probes are all blue; the grounds are all black. All remaining probes (32 for the 32-Channel Logic Pod, and 4 for the 4-Channel Logic Pod) have grey caps.

For the 32-Channel Logic Pod, the probe color coding is shown in Figure 3 and on the Logic Pod Color Codes screen (accessed via the Special Functions screen by depressing S. FUNC, then "2"); for the 4-Channel Logic Pod, the color codes are shown in Sections II.A.2 below and Section V.B.1. The data probes of the 32-Channel Logic Pod use the standard electronic color code, with the tip being the most significant digit and the wire being the least significant.

The header pin assignments are shown below; for the 32-Channel Logic Pod, P1B is the 20-pin header for Probes 0-7, and P1A is the 26-pin header for Probes 8-31. Note that the numbering on the connector receptacle does **not** match that on the mating connector. All pin numbers in this manual refer to the connector receptacle. (The connector is housed within the pod.)



CONNECTOR RECEPTACLES ON END OF 32-CHANNEL LOGIC POD

1. 32-CHANNEL LOGIC POD PROBES

The probe set with 16 wires is called the 8-Channel Probe Set; it contains the Clock Probe, the Clock Qualifier Probe, Probes 0-7 and six grounds. (This group is equipped for high-speed recording.) The probe set with 26 wires is called the 24-Channel Probe Set, and contains Probes 8-31 and two grounds.

This pod contains a potentiometer and a switch for selecting either "Fixed" or "Variable" voltage threshold. The "Fixed" position sets the voltage threshold at TTL levels (1.4 V). The "Variable" position allows setting the voltage threshold between -9 V and +9 V by turning the adjusting screw. To measure the voltage, use the reference line that is (internally) connected to Pin 19 on the pod's 20-pin connector, and use a high-impedance meter. The impedance of this node is about 1 Kohm.

Probe color codes are shown in Figure 3.

2. 4-CHANNEL LOGIC POD PROBES

The 4-Channel (Glitch-Capture) Logic Pod has six probes: four for data, one for an external clock, and one for clock qualification. (Clock qualification functions on this pod the same as it does on the 32-Channel Logic Pod, LP-320). Each probe consists of two microhooks attached to a length of twisted-pair cable. The all-black hook in each pair is a ground probe, and should be connected to a circuit ground local to the signal being observed. The remaining six hooks are color coded as follows:

	BODY (tip)	PLUNGER (cap)
External Clock	Red	Red
Clock Qualifier	Blue	Blue
Channel 0	Grey	Black
Channel 1	Grey	Brown
Channel 2	Grey	Red
Channel 3	Grey	Orange

3. CONNECTING LOGIC POD TO SYSTEM UNDER TEST

To connect a logic pod to the circuit under test, insert the pod's cable connector plug into the connector receptacle on the front panel. Then connect the desired number of probes (which corresponds to the number of selected input channels) to the points on the circuit to be tested.

To sample data synchronously with an external clock, connect probes for the clock and (optionally) the clock qualifier to the circuit. The input buffers for clock, clock qualifier, and Data Channels 0-3 have been designed to be very fast, and care must be taken to minimize coupled noise.

WARNING: THE CLOCK AND CLOCK QUALIFIER WIRES SHOULD ALWAYS BE TWISTED WITH A CONNECTED GROUND LEAD. FOR OPERATION ABOVE 10 MHz, THE LOWER DATA CHANNELS SHOULD ALSO BE TWISTED WITH THE GROUND LEADS PROVIDED.

B. CONNECTION VIA MICROPROCESSOR PODS

WARNING: DO NOT PLUG THE POD CONNECTOR INTO, OR UNPLUG IT FROM, THE FRONT-PANEL CONNECTOR RECEPTACLE WHILE THE ML4100 IS POWERED UP.

To connect a microprocessor pod, insert its cable connector plug into the connector receptacle on the front panel, then clip the probe to the microprocessor; the brown wire on the colored ribbon cable connects to Pin 1 of the microprocessor. Information obtained internally from this probe will be used by the ML4100 to determine some of its setup parameters.

For setup procedures, see Section III.H, (Display) Format Setup Screen. For trigger configuration details and status codes, see Section VI, Microprocessor Pods.

C. CONNECTION VIA STD BUS POD (8P-080)

WARNING: DO NOT PLUG THE POD CONNECTOR INTO, OR UNPLUG IT FROM, THE FRONT-PANEL CONNECTOR RECEPTACLE WHILE THE UNIT IS POWERED UP.

The STD Bus Pod consists of a circuit board and a cable for connection to the ML4100.

To connect the STD Bus Pod, attach its cable connector plug to the connector receptacle on the front panel; then insert the circuit card into an unused slot in the STD bus backplane of the unit under test, or install it as an extender between the bus and a card on the bus.

For operating details, see Section VII, STD Bus Pod.

D. ROM EMULATOR MODULE (RE-016) CONNECTION

WARNING: DO NOT CONNECT OR DISCONNECT THE ROM EMULATOR MODULE WHILE THE ML4100 IS POWERED UP.

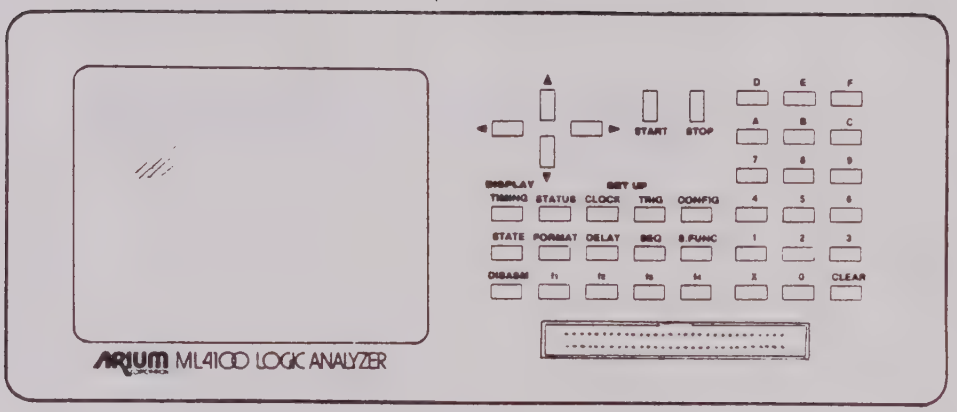
To connect the ROM Emulator Module, insert the module's cable connector into the P2/ROM EMULATOR connector receptacle on the back panel. Then connect the proper DIP plug directly between the ROM sockets of the system under test and the ROM Emulator pod.

For operating details, see Section VIII, ROM Emulator Module.

(Blank)

III. DISPLAY AND SETUP OF OPERATING PARAMETERS

(Figure 1, repeated)
FRONT PANEL



A. GENERAL INFORMATION RE ALL SETUP AND DISPLAY SCREENS

To access the ML4100 for operation, use the front-panel setup keys: STATUS, CLOCK, CONFIG, TRIG, SEQ, DELAY, and FORMAT. Each of these keys calls up a CRT screen offering parameter selections and/or setup information. Changeable fields on the CRT display are reverse-videoed and the cursor position blinks; the cursor automatically skips over nonchangeable fields.

Each screen contains complete prompting information about how to set up each parameter, including exactly what the preset options are, the specific hexadecimal key designated to select each option, and the allowable ranges of parameter values. Also, the system protects against most wrong entries and selections by not accepting them, and beeping to warn the user.

As soon as a parameter has been set, any other parameter dependent upon it is updated and displayed, usually immediately. (In some cases, a dependent parameter is not updated until the cursor has left the field of the parameter being changed.)

To select or change any setup (operating) or display parameter, move the cursor to the parameter's field, which will cause it to blink, indicating its readiness to be set up. Each parameter may then be set in one of two ways: either there is a limited number (usually 2 or 3) of preset options which may be selected by the user, or the actual value of the parameter (usually numeric) may be entered by the user. Both operations are performed directly via the keys on the hexadecimal keypad, which contains a key for each digit from 0-9, a key for each letter from A-F, a CLEAR key, and an "X" key. (See Figure 1.)

While the ML4100 is recording, a blinking "ACTIVE" message is displayed in the upper right of the six setup screens; during this time, setup parameters may be viewed on each screen, but not changed. Recording must be stopped (by depressin- STOP) before any setup parameter may be changed.

B. (SYSTEM) STATUS SETUP SCREEN (VIA "STATUS" KEY) (See Fig. 4)

The (System) Status screen acts as a base of operations for the ML4100 user. It is a passive screen which only presents information, as compared with the six setup screens (Clock, Configuration, Trigger, Sequence, Delay and Format), which not only present information but also are interactive screens via which the user can select and change operating parameters.

The (System) Status screen gives an overall view of the six setup screens by showing how the ML4100 is currently set up, i.e., what major parameters have been selected (or defaulted to) on each of the setup screens. It is often more useful to view all these major parameters at once on this display, rather than to view them a few at a time as they are shown on each individual setup screen.

However, the user will want to move to the individual setup screens to view the setup parameters in greater detail, and the user must move to these other screens to select and change the setup parameters. An asterisk preceding a Status screen label denotes a key which accesses such a setup screen.

To view overall major system parameters on the (System) Status setup screen, depress the STATUS key on the front panel.

To view system parameters in greater detail than is shown on the (System) Status setup screen, to set up the ML4100 for initial operation, and/or to change any detailed system parameters while using the ML4100, use the following setup keys on the front panel:

CLOCK	Recording/sampling rate and clock qualification
CONFIG (Configuration)	Recording width/number of channels and data qualification
TRIG (Trigger) SEQ (Trigger Sequence) >	Triggering condition, which stops the recording process
DELAY (Trigger Delay)	
FORMAT (or use default format)	Data display format

Refer to Figure 4, (System) Status Setup Screen, for an example of how one particular set of selections will be displayed.

Figure 4

(SYSTEM) STATUS SETUP SCREEN

```
STATUS      HB  c 1983, ARIUM Corp.
*CLOCK                  *CONFIGURATION
  Source : External       Width     : 32  bits
  Sense  : rising         Data Qual:   off
  Qual   : always

*DELAY   : 00150 Clocks
*SEQUENCE:  A then B then (C without D)

*TRIGGER:                *FORMAT POD:  8085

     BIN  STS  ADR DAT
A: XXXX XXXX 003C  XX
B: XXXX XXXX 009C  XX
C: XXXX XXXX 009F  XX
D: XXXX XXXX 00A0  XX
```

NOTE: An asterisk preceding a Status screen label denotes a key which accesses that particular parameter.

Following are the system parameters as they are displayed on the (System) Status screen and the selections for each parameter available to the user.

1. CLOCK

Source:	External
Sense:	Rising or Falling (Edge)
Qual: (Clock qualification):	Always, Low or High
	--OR--
Source:	Internal
Period:	_____ (4-digit numeric value and units)

2. CONFIGURATION

Width:	32, 16, 8, or 4 bits
Data Qual:	Off
	--OR--
Width:	32, 16, 8, or 4 bits
Data Qual:	Comb.
Record:	Only or All but
Occur. of:	None or _____ (Trigger words)
	--OR--
Width:	32, 16, 8, or 4 bits
Data Qual:	State
Enable:	None or _____ (Trigger words)
Disable:	None or _____ (Trigger words)

3. (TRIGGER) DELAY: _____ Clocks (5-digit value) (Default = 00000)

4. (TRIGGER) SEQUENCE: _____

Displays the sequence from the 19 possible shown in Figure 8, Trigger Sequence setup screen, or "User-def. seq.", as shown in Figure 9.

5. TRIGGER:

A: XXXX 0011 0100 XX
B: XXXX 0011 0200 XX
C: XXXX XXXX XXXX XX
D: XXXX XXXX XXXX XX

Displays the selected trigger words in number base and the selected or default (State) display format. Default is all "Xs". In this example, trigger words A and B are set to match "fetches" at addresses 0100H and 0200H if an Intel pod is attached.

6. **FORMAT:** Pod: _____ (mode)

Most pods have more than one mode, in which case the f1, f2, f3, or f4 key is used to select the particular mode via the (Display) Format setup screen; e.g., the f1 key would select the 8085 mode of the 8I-080 microprocessor pod.

C. CLOCK SETUP SCREEN (VIA "CLOCK" KEY) (See Figure 5)

Figure 5

CLOCK SETUP SCREEN

```
                    CLOCK

SOURCE (0=internal,1=external): External

INTERNAL:
 Clock period (10 ns - 200 ms): 0010  nsec
   (f1=nsec, f2=usec, f3=msec)
 Clock rate                   : 12.5 MHz
 Record time                  : 80.0 usecs

EXTERNAL:
 Clock edge (0=fall,1=rise)   : rising
 Qualifier valid              : always
    (0=low, 1=high, 2=always)
```

To access the Clock setup screen, depress the CLOCK key. This screen displays the internal and external clocking parameters and is used to select and change specific parameters.

1. SELECTING CLOCK SOURCE

An internal clock is used when analyzing problems related to logic, hardware, and timing, and when measuring time. An external clock is used for analyzing bus problems and software, and occurrences synchronous to the user's system. External clock rates may not exceed 70 MHz (4-bit mode). (For rate limits in other modes, see the table in Section III.D.1.)

To select (clock) Source, move the cursor to the field labeled "Source", then select either "Internal" by depressing the 0 key on the keypad, or "External" by depressing the 1 key on the keypad. An internal clock uses time periods measured in nanoseconds (nsec), microseconds (usec), or milliseconds (msec).

2. USING AN INTERNAL CLOCK

To set the clock period (from 10 nsec to 200 msec), move the cursor to the first field to the right of "Clock period" and enter the desired value via the hexadecimal keypad; then move the cursor to the second field and select one of the three preset options for units by depressing f1 (nsec), f2 (usec), or f3 (msec). The corresponding clock (recording) rate, or frequency, and record time will be automatically displayed as soon as the cursor leaves the Clock Period numerical field.

Not every frequency is available; when an unavailable selection is made, the closest available clock frequency is used.

3. USING AN EXTERNAL CLOCK

The clock detection circuitry of both the LP-320 and the LP-040 is very high speed. To avoid extra clocks due to stray pickup, a connected ground lead should always be twisted with the clock lead. (The LP-040 leads are twisted pairs, but the LP-320 leads are not.) Failure to twist a connected ground lead with the clock lead is likely to produce extra ("double") clocking; if these clocks occur at a period of less than the minimum, the logic analyzer may function incorrectly.

Because of the ML4100's internal configuration, a minimum of 17 clocks must occur after the trigger condition for correct operation. These 17 clocks are automatically compensated for in all displays; the user need not be concerned about them unless the external clock stops within 17 clocks after the trigger condition.

a. Clock Edge

To select the Clock Edge (the edge of the clock at which data is to be sampled), depress the 0 key for "Falling" or the 1 key for "Rising". "Falling" means that the data is sampled on the high-voltage-to-low-voltage transition of the external clock input, as it crosses the threshold voltage; "Rising" means that data is sampled on the upward transition.

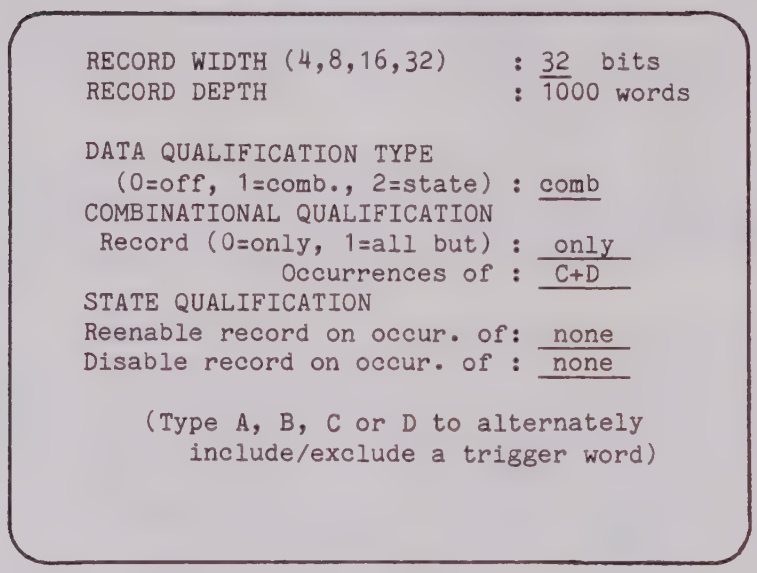
b. Clock Qualifier

To select the clock qualifier, depress 0 (Low), 1 (High), or 2 (Always) on the keypad. "Low" means that the clock qualifier is valid only when the qualifier input is low; "High" means that it is valid only when the input is high; and "Always" means that it is always valid. (External clocks are recognized only when the qualifier is valid.)

NOTE: The clock qualifier has no effect with an internal clock, and cannot be used with microprocessor pods.

D. CONFIGURATION SETUP SCREEN (VIA "CONFIG" KEY) (See Figure 6)

Figure 6
CONFIGURATION SETUP SCREEN


RECORD WIDTH (4,8,16,32) : 32 bits
RECORD DEPTH : 1000 words
DATA QUALIFICATION TYPE
(0=off, 1=comb., 2=state) : comb
COMBINATIONAL QUALIFICATION
Record (0=only, 1=all but) : only
Occurrences of : C+D
STATE QUALIFICATION
Reenable record on occur. of: none
Disable record on occur. of : none
(Type A, B, C or D to alternately
include/exclude a trigger word)

To access the Configuration setup screen, depress the CONFIG key.

1. RECORD WIDTH

To select Record Width, move the cursor to the field labeled "Record Width" and enter the number of bits (channels) (4, 8, 16, or 32) via the key-pad. (If any other value is entered, it will not be accepted, and the next highest of these four values will be displayed.) The record (word) depth available memory) will then be automatically displayed. (A wider word has less depth; e.g., 8000 samples per channel will be recorded in the 4-channel mode, and 1000 samples per channel in the 32-channel mode.)

The maximum clock rate (speed) for each record width varies from 100 MHz (for 4-bit) to 12.5 MHz (for 32-bit). The following chart shows the relationships between record width, clock rate (speed), and record depth.

Record Width	Maximum Clock Speed	Record Depth
4 bits (int. clock)	100 MHz	8000 samples/words
4 bits (ext. clock)	70 MHz	8000 samples/words
8 bits	50 MHz	4000 samples/words
16 bits	25 MHz	2000 samples/words
32 bits	12.5 MHz	1000 samples/words

2. DATA QUALIFICATION (in the 32-channel/bit mode only)

Data qualification is used to record only desired pieces of data to conserve memory and to make problem areas more visible. It permits determination on an individual (combinational) or sequential (state) basis as to whether or not any given word (data sample) will be recorded; it is available in 32-channel mode only. Note that data qualification does not affect recognition of trigger words, but does affect the trigger delay; i.e., a word included in the trigger sequence will be recognized regardless of whether or not data qualification allows it to be recorded, but only qualified words are counted in the trigger delay.

To select the type of Data Qualification, enter a 0 (Off), 1 (Combinational), or 2 (State) on the keypad.

To select the trigger words to be used as data qualifiers, use the A-D keys. When more than one word is entered, the display will automatically show a "+" between words, meaning "or". Each A-D key toggles on/off when depressed repeatedly, causing the character to alternately appear and disappear.

a. Combinational Data Qualification

To select a combinational data qualifier, depress 0 (Only) or 1 (All but) in the field labeled "Record", then enter A, B, C, and/or D in the "Occurrence of" field to indicate the trigger words upon which the selected qualifier will act. All four trigger words (A-D) may be used. (Setting of trigger words is described in Section III.E.)

For example, to record only program fetches in the area of 0100H to 02FFH when using a microprocessor pod, proceed as follows. First set Trigger Word C to fetches with the Address field 01XX and the Data field XX. Set Trigger Word D similarly, with 02XX in the Address field. Set Data Qualification Type to "comb"; then under Combinational Qualification set "Record" to "only" and set "Occurrences of" to "C+D". (This setup is depicted in Figure 6.)

b. State Data Qualification

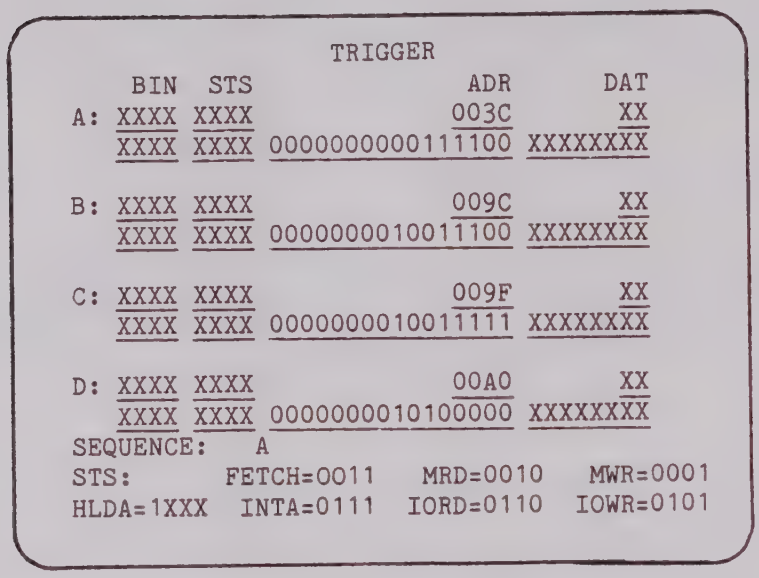
To select a state data qualifier, enter the selected trigger conditions upon which recording will be enabled/or disabled in the appropriate fields. Recording always begins enabled when the START key is depressed. Only one or two trigger words per field may be used; although the field is large enough to enter three or four words, the logic analyzer will not start recording data if more than two have been entered.

NOTE: When state data qualification is being used, trigger word "D" is not available for trigger sequence and, if either field ("Reenable..." or "Disable...") contains two words, then trigger word "C" is also unavailable for trigger sequence. (It is convenient to use "C+D" for data qualification, so as to leave A and B open for triggering.)

For example, with state qualification, to record everything from the time an instruction is fetched at 3F27H until an instruction is fetched at 4010H, set trigger words C and D to fetches at those locations, respectively, enable state qualification, and then set "Reenable..." to C and "Disable..." to D.

E. TRIGGER SETUP SCREEN (VIA "TRIG" KEY) (See Figure 7)

Figure 7
TRIGGER SETUP SCREEN



To access the Trigger setup screen, depress the TRIG key. The TRIG key calls up the Trigger setup screen with which four trigger words (events), labeled "A", "B", "C", and "D", are displayed and defined. On the display, there are two lines for each trigger word. The word appears in the number base selected (via the State portion of the Display Format screen)-- binary, octal, decimal, or hexadecimal (but not ASCII)--on the upper line, and in binary on the lower line. (When specifying the number base via the Display Format screen, use the State fields.) For the user's convenience, the trigger sequence is also displayed, in a format identical to the State format.

Fields are displayed from left to right, in ascending numerical order (using the numerals labeling the fields on the Format screen); "off" fields are not displayed. All numbers are displayed with the least significant digit at the right of the field; for example, in a binary field of Probes 3-0, Bit 0 is at the right of the field.

To define a trigger word, move the cursor to either line and use the keypad to enter the appropriate value; the corresponding value on the other line will be automatically displayed. To change any character in a trigger word, just move the cursor to that character and enter the new character on the keypad.

With a logic pod, the default number base for the upper line is binary, and is so labeled on the screen. With microprocessor and bus pods, the default number base is hex for many fields, but is determined by the particular pod and its requirements. Microprocessor pods also label fields as "STS" (status bits), "ADR" (address), and "DAT" (data).

An attempt to enter a digit which is too large for the selected number base will be rejected, and will cause the unit to beep once as an error warning. In decimal number base, as further protection against entering unacceptable values, the ML4100 will truncate the upper bits of a value that is larger than allowed; e.g., in decimal number base, if "18" is entered in a 4-bit field, as soon as the cursor leaves the field, the lower 4 bits are used, leaving a "2", an acceptable value.

To change the number base from the default or previously set base, use the Display Format screen via the FORMAT key. (See Section III.H.) To indicate a "Don't Care" when entering a trigger word on either line, depress "X" on the keypad. Any resulting indeterminate character on the upper line will be displayed as a "?". In decimal number base, a "Don't Care" may not be entered for an individual digit because of the binary nature of the unit.

The bottom two lines of this screen may contain status code prompts (information about the status fields). The presence of such information depends upon which microprocessor pod is being used and whether the appropriate disassembler has been installed. Possible **status code prompts** are:

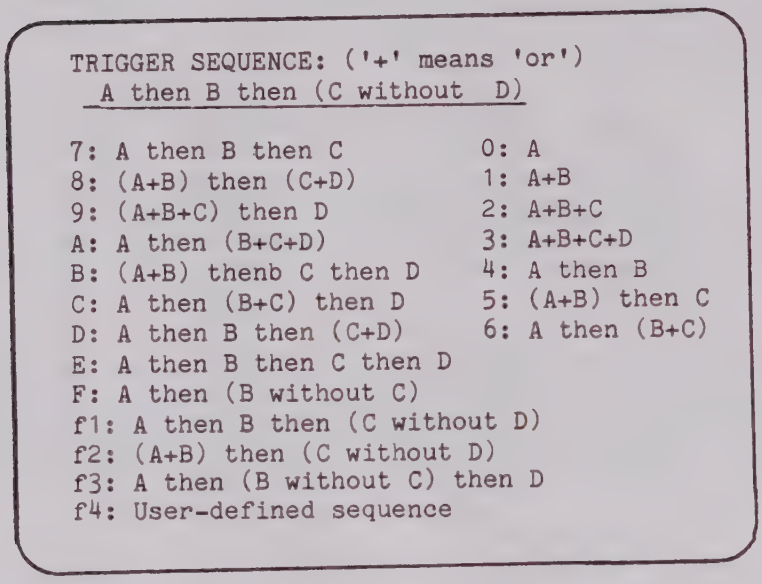
Prompt	Status Code
FETCH	Fetch
MRD	Memory Read
MWR	Memory Write
INTA	Interrupt Acknowledgment
IORD	I/O Read
IOWR	I/O Write

(For further status code details, see Section VI, Microprocessor Pods.)

Figure 7 shows four trigger words defined to be matched with any type of access (Fetch, Read, Write, etc.) to one of four addreses. In all four cases shown, the trigger condition is satisfied no matter what data is transferred during that cycle (Data XX).

F. TRIGGER SEQUENCE SETUP SCREEN (VIA "SEQ" KEY) (See Figure 8)

Figure 8
TRIGGER SEQUENCE SETUP SCREEN


TRIGGER SEQUENCE: ('+' means 'or')
A then B then (C without D)
7: A then B then C
8: (A+B) then (C+D)
9: (A+B+C) then D
A: A then (B+C+D)
B: (A+B) thenb C then D
C: A then (B+C) then D
D: A then B then (C+D)
E: A then B then C then D
F: A then (B without C)
f1: A then B then (C without D)
f2: (A+B) then (C without D)
f3: A then (B without C) then D
f4: User-defined sequence
0: A
1: A+B
2: A+B+C
3: A+B+C+D
4: A then B
5: (A+B) then C
6: A then (B+C)

To access the Trigger Sequence setup screen, depress the SEQ key. The trigger sequence defines the trigger condition which will stop the recording of data after the selected trigger delay. If the delay is zero, recording stops immediately upon occurrence of the trigger condition. (See Section III.G, Trigger Delay, and Figure 10).

A trigger sequence condition is formed by various combinations or sequences of the selected trigger words (events), up to four words (labeled A, B, C, and D) and four levels. (Trigger words are defined via the Trigger setup screen; see Section III.E and Figure 7).

The Trigger Sequence setup screen is used to select (and to change the selection of) any of the 19 predefined trigger sequences, and to access (via the f4 key) the User-Defined Sequence setup screen (see Section III.F.2, below).

1. PREDEFINED TRIGGER SEQUENCES (See Figure 8)

The Trigger Sequence setup screen displays the 19 predefined trigger sequences, along with the corresponding keypad entries for selecting them.

To select or change a predefined trigger sequence, depress the key which corresponds to the selected sequence; it will then be displayed on the second line of the screen, and may be changed via the keypad. The default is selection "0", predefined as "A".

Other selections include combinations or sequences of trigger words (events) to make up the trigger condition. For example, the sequence designated "7" ("A then B then C") will cause recording to continue until trigger word "A" is encountered, then until trigger word "B" is encountered, then until trigger word "C" is encountered. At this point, the entire trigger condition has been satisfied, and the delay will begin. (Recording will stop immediately if the delay is zero.)

Sequence f1 ("A then B then (C without D)") is identical to Sequence 7, except that, if "D" is encountered after "B" but before "C", then the ML4100 will begin again to search for "A". This sequence can be very useful when it is desired to trigger only when one event occurs immediately after another. To do this, use a sequence such as the f1 sequence, with "D" containing all "Don't Cares"; in this case, "B" and "C" will have to occur with no other cycles in between to satisfy the trigger condition.

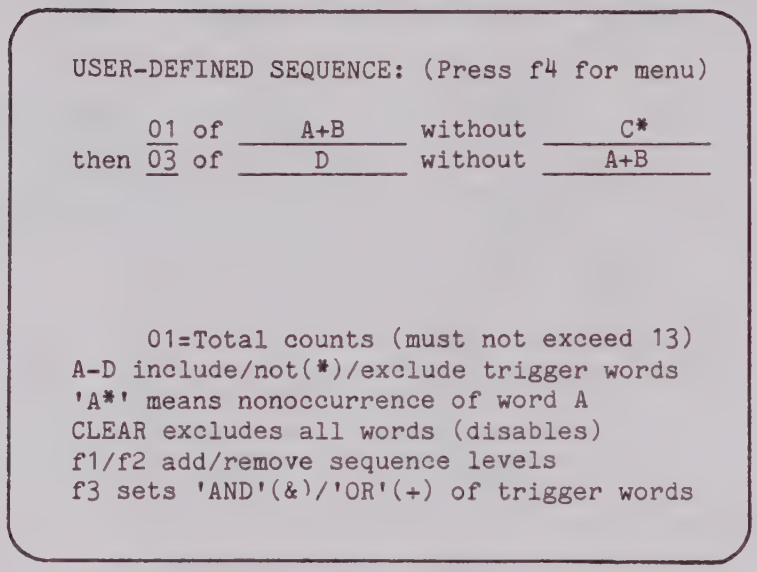
Only in the 32-channel operation are all 19 trigger sequences (including the user-defined sequence) available, as shown in the table below.

Recording Width	Available Trigger Sequences
4 channels	0, 1, 2, and 3
8 channels	0, 1, 2, and 3
16 channels	0, 1, 4, and f4*
32 channels	All (including user-defined)

* User-defined sequences

2. USER-DEFINED TRIGGER SEQUENCE (See Figure 9)

Figure 9
USER-DEFINED SEQUENCE SETUP SCREEN


USER-DEFINED SEQUENCE: (Press f4 for menu)
01 of A+B without C*
then 03 of D without A+B
01=Total counts (must not exceed 13)
A-D include/not(*)/exclude trigger words
'A*' means nonoccurrence of word A
CLEAR excludes all words (disables)
f1/f2 add/remove sequence levels
f3 sets 'AND'(&)/'OR'(+) of trigger words

To access the User-Defined Sequence setup screen for selecting or changing the user-defined trigger sequence, depress the f4 key (while viewing the Trigger Sequence setup screen). This brings up a new screen ("User-Defined Sequence") and presents a complex sequence format with one to four levels. Each level displays three fields: a 2-digit field for number of counts for that level, an 11-character field for a trigger word combination, and a second 11-character field for a "without" trigger word combination.

When recording, the ML4100 waits at each level until that condition has been satisfied, then proceeds to the next level. When the last level has been satisfied, the trigger condition is satisfied. If the entered count for a level is more than 1, then that level must be satisfied that number of times before the ML4100 proceeds to the next level. If the "without" condition is encountered at a level, then the whole sequence starts over at the first level.

This screen's default format displays only the first trigger sequence level. To add a trigger sequence level, depress f1; to remove the lowest level, depress f2.

To enter the number of counts for each level, enter the value via the keypad. Up to 13 total counts (among all 4 levels) are allowed. For example, if a total of 9 counts has been entered into the first three levels, then no more than 4 counts may be entered into the fourth level. After entry of the number of counts for a level, the new total number of counts is displayed as soon as the cursor is moved. Any level whose count is greater than 13 will be limited to 13 if its count field is exited. If the new total number of counts exceeds 13, any attempt to leave the screen causes the ML4100 to beep 3 times, flashes the "must not exceed 13" message, and does not allow leaving the screen.

To enter the trigger word combinations for each level, use the keypad to enter the desired combination of available trigger words ("A", "B", "C", and "D"). To include a trigger word in a sequence level, depress the appropriate key one time. (Take care to depress the key quickly and surely; do not linger on the key, or it will repeat and change the entry.) If a word has already been entered in the field, the new word will be entered as an "or" word (shown as "+") or an "and" word (shown as "&"), depending upon the mode. For example, in the "or" mode, if "A" is already in the field, depressing the B key will enter "+B", and the field will show "A+B", meaning "A or B".

To change an entered trigger word to a "not" condition, depress the same key a second time (with the cursor in the appropriate field), and the display will place a "*" ("not") on its right. E.g., if "A+B" is already in the field, depressing the B key will add an asterisk to the "B", and the field will show "A+B*", meaning "A or not B".

To exclude an entered trigger word, depress the same key a third time and the word (and its attached asterisk) will disappear from the field display. If the word was not a negative (i.e., had no asterisk) the key must be depressed twice toggling through the negative condition to get to the excluded one. Depressing the same key a fourth time will toggle back to include the trigger word, etc., repeating the pattern of inclusion/negation/exclusion endlessly.

To change (all of) the "+" ("or") characters in the trigger word to "&" ("and"), depress the f3 key; depressing f3 again toggles all the "&" characters back to "+" characters, in another repeating pattern.

To exclude (erase) an entire trigger word sequence from a field, depress CLEAR, and the display will show "(disabled)" in the field.

To return to the Trigger Sequence setup screen, depress the f4 key, which toggles the display back and forth between the Trigger Sequence and the User-Defined Sequence setup screens.

NOTE: When setting a user-defined sequence, be sure to exit from the User-Defined Sequence setup screen directly to any screen except the Trigger Sequence setup screen. Exiting from the User-Defined Sequence setup screen directly to the Trigger Sequence setup screen will reset the Trigger Sequence to its default condition (Sequence 0, which is "A"); the user-defined sequence will be remembered, but will not be used to trigger the analyzer.

To collect data without the possibility of a trigger condition stopping the recording of data (i.e., to collect data indefinitely until the STOP key is depressed), set the user-defined trigger sequence to exclude all trigger words by entering "01" as the number of counts, then depressing CLEAR to enter "disabled" in both trigger word fields. The display will then show "01 of (disabled) without (disabled)". (This works in the 32- and 16-channel modes.)

G. TRIGGER DELAY SETUP SCREEN (VIA "DELAY" KEY) (See Figure 10)

To access the Trigger Delay setup screen, depress the DELAY key.

To delay the stopping of recording ("storage") of data past the occurrence of the trigger condition, use the keypad to enter (in the top field) the number of qualified clocks after the trigger condition occurrence at which recording (storage) should stop. This top-field value is the "trigger delay".

Or, to begin recording at a selected number of clocks after or before the trigger condition occurrence, enter that number of clocks in the second field, then select "A" (after) or "B" (before) in the third field. (A sufficiently long trigger delay will place the entire recording interval after the trigger condition.)

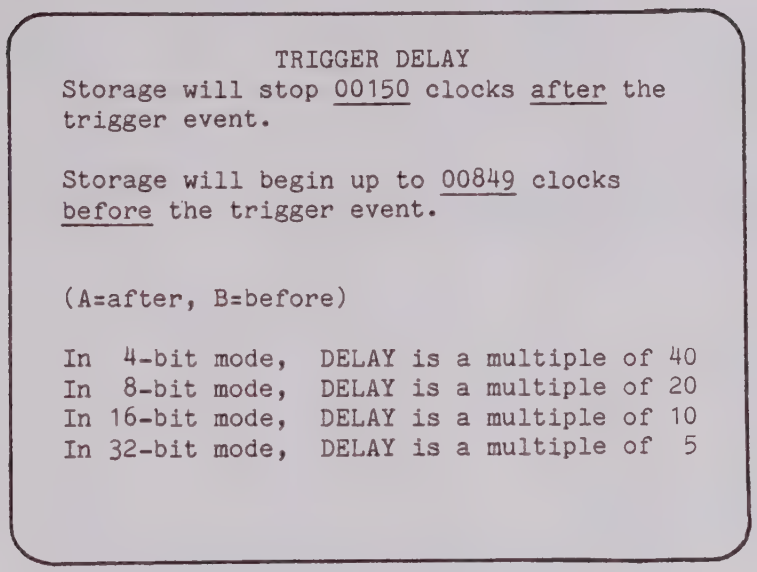
Only one of the two clock-number fields need to be entered; the corresponding number in the other number field will be automatically displayed. (The difference between these two numbers is the record depth.) The delay period includes only clock-qualified and data-qualified clocks.

NOTE: Because of the ML4100's internal configuration, a minimum of 17 clocks must occur after the trigger condition for correct operation. These 17 clocks are automatically accounted for in all displays; the user need not be concerned about them unless the external clock stops within 17 clocks of the trigger event. Only two clocks are required if the STOP key is depressed (Rev H software and later only).

NOTE: Entering a number in the second field is only an alternate means of setting the trigger delay (the delay from the trigger event until stopping of the recording process). Trigger delay forces a number of data words to be stored after the trigger event. It cannot, however, guarantee the number of words before the trigger event. For example, if delay is set to 200, but the trigger condition is satisfied on the tenth clock, then there will be only ten data words prior to the trigger.

Sometimes this makes it difficult to capture the beginning of a software loop. This can be overcome by triggering on a second occurrence of a particular event. For example, trigger words "A and B" could be set to the same thing and a sequence of "A then B" set into the machine. Alternatively, a user-defined sequence of 13 occurrences of A would cause a minimum of 12 full copies of the loop to be captured prior to the trigger.

Figure 10
TRIGGER DELAY SETUP SCREEN


TRIGGER DELAY
Storage will stop 00150 clocks after the trigger event.
Storage will begin up to 00849 clocks before the trigger event.
(A=after, B=before)
In 4-bit mode, DELAY is a multiple of 40
In 8-bit mode, DELAY is a multiple of 20
In 16-bit mode, DELAY is a multiple of 10
In 32-bit mode, DELAY is a multiple of 5

H. DISPLAY FORMAT SETUP SCREEN VIA "FORMAT" KEY (Figure 11)

To access the Display Format setup screen, depress the FORMAT key. This screen is used to view and select formats (bit groupings and number base) for displaying data on the Timing, State, Disassembly, and Trigger screens. (The Trigger setup screen uses the same format setup as does the State display screen, and so does not appear separately defined on this Format screen.) Data may be displayed in any of several number bases: binary, octal, decimal, hexadecimal, or (except for the Trigger screen) equivalent ASCII values. Data may be reformatted or redisplayed in any of the display modes without recapture (re-recording); this is especially convenient for displays in which a different base or bit grouping is needed.

Figure 11
DISPLAY FORMAT SETUP SCREEN

```
                    DISPLAY FORMAT

TIMING:
Line     1  2  3  4  5  6  7  8  9 10 11 12
Probe   28 29 30 31 XX XX XX XX XX XX XX XX

STATE:          STS ADR DAT
Field         1   2   3   4   5   6   7   8
Base        BIN BIN HEX HEX OFF OFF OFF OFF
High Probe   31  27  23  07  00  00  00  00
 Low Probe   28  24  08  00  00  00  00  00
Data Sense  NON NON NON NON NON NON NON NON
Bases: A=ASC B=BIN C=OCT D=DEC E=HEX F=OFF
Senses: 0=NON (non-inverted), 1=INV

POD MODE (also sets defaults):  8085
f1= 8085   f2=8035/9  f3=8031/2
```

1. FORMAT FOR TIMING DISPLAY MODE (See Figure 11)

On the Display Format setup screen, the timing display assignments appear at the top, with up to 12 timing lines identified. Each field from (left to right) corresponds to a timing display position (from top to bottom) on the timing screen. The same probe may be displayed on more than one line, and blank ("XX") lines may be used for spacing.

To define or change a timing line, enter the probe number via the keypad. (When a probe number is defined, its timing line will be automatically displayed on the Timing display screen; when a probe number is changed, the line will be automatically relabeled.) For example, to display Probes 0-3 on the timing display, set the first four fields on the format screen to 0, 1, 2, and 3, respectively.

To disable display of a line, depress the "X" key, and "XX" will be displayed in the Probe field. (This will also automatically blank out that line on the TIMING display screen.)

2. FORMAT FOR STATE DISPLAY MODE (See Figure 11)

The state display format parameters appear in the middle of the Display Format setup screen. There are positions for describing eight fields which will correspond to up to eight fields (columns) on the state and timing screens. The defined fields will appear left to right in ascending numerical order on the state and trigger screens. All fields are displayed with the least significant digit on the right. For example, in binary fields, Bit 0 is in the rightmost position.

For each of 8 fields of data, there are 4 selections: (number) Base, High Probe, Low Probe and Data Sense. If the unit is connected to a microprocessor, Fields 2, 3, and 4 are typically labeled "STS" (Status), "ADR" (Address), and "DAT" (Data), respectively. "ADR" and "DAT" are set to "Hex" for the hexadecimal display on the State screen.

Data may be displayed in more than one field, and fields may overlap. Fields may be of any width, up to the limit of the number of probes or of space on the CRT display. Each field may contain a maximum of 22 bits.

To select number Base, use the keypad to enter "A" (ASCII), "B" (Binary), "C" (Octal), "D" (Decimal), "E" (Hexadecimal), or "F" (Off, which disables the field). (This will automatically change the data line for this field on the State display screen and on the Trigger setup screen.)

To enter Probe Range for each field, use the keypad to enter the high probe number and the low probe number, inclusive. The probe numbers included in the field will be consecutive, and the high probe number must not be less than the low probe number. These probe numbers (on the logic pods) refer to the input data channel numbers.

To select Data Sense, use the keypad to enter "0" (Noninverted) or "1" (Inverted). Any field may be inverted. Inverted fields will display the 1's complement of the data in the selected number base. Inversion does not affect the Timing display, and there is no provision for inverting timing lines. Inversion does affect the values on the Trigger display screen.

(The Timing and State displays may comprise entirely different combinations of probes.)

3. DISPLAY FORMAT FOR DISASSEMBLY (VIA MICROPROCESSOR PODS) AND OTHER MODES (See Figures 11 and 12)

At the bottom of the Display Format screen is the pod mode choice, which sets default parameters according to the type of pod, including the Disassembly display mode, which allows decoding of microprocessor instructions ("reverse assembly").

To select a particular pod mode, cursor to the bottom field and depress the appropriate key (f1, f2, f3, or f4), according to the prompt displayed on the bottom line.

(See Figure 12 for currently available pod mods.)

Figure 12
POD MODE DESIGNATIONS

Pod Catalog Number	Designation of Mode	Supports:
8I-080	8031/2	8031, 8032
	8035/9	8035, 8039, 8040
	8085	8085
8M-080	6800	6800
	6802/8	6802, 6808
8M-089	6809	6809
	6809E	6809E
8R-065	6502	6502
	6512	6512
	65C112	65C112
	65CX02	65C02, 65C102
8Z-080	Z80	Z80, Z80A, Z80B, Z80C
	Z80REF	Z80, Z80A, Z80B, Z80C (all with refresh recorded)
8P-080	STDBUS	STD bus bus controls
	STDINT	STD bus interrupt controls
	STD24B	STD bus with 24-bit addressing
LP-320	LOGIC4	4-channel operation
	LOGIC8	8-channel operation
	LOGC16	16-channel operation
	LOGC32	32-channel operation
LP-040	GLITCH	With glitch capture
	NOGLCH	Without glitch capture

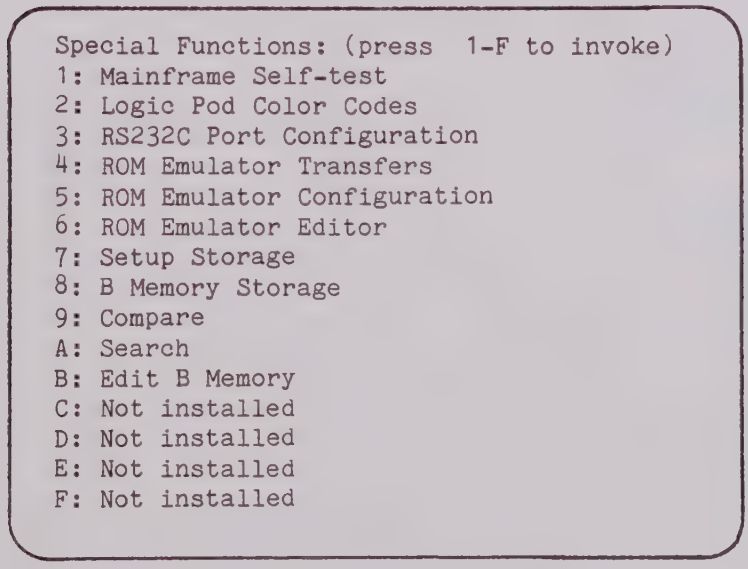
I. SPECIAL FUNCTIONS SETUP SCREEN (VIA "S. FUNC" KEY) (See Figure 13)

The ML4100 provides for up to 15 special functions (labeled "1" through "F"), of which some are standard and some are optional. To access the Special Functions setup screen and display a listing of the special functions, depress the S. FUNC key. Then, to select a particular special function, use the keypad to depress the hexadecimal character which corresponds to the function desired.

There are two standard special functions which are installed on all units: Mainframe Self-test (accessed by depressing the 1 key) and Logic Pod Color Codes (accessed via the 2 key). Other special functions are optional; if any has been installed on a unit, it will be listed by description on the Special Functions screen. Any optional special functions which are not installed on a unit will be listed on this screen as "Not installed".

The Special Functions setup screen will show some or all of the following screens, depending upon which options are installed in the ML4100. In the example below, the ML4100 has both an RE-016 ROM Emulator Module (Special Functions 3-6) and the AB-032 Setup/Data Memory Option (Special Functions 7-B) installed.

Figure 13
SPECIAL FUNCTIONS



1. MAINFRAME SELF-TEST (Standard)

The Mainframe Self-test includes:

ROM Checksum
RAM Test
Trace Address
Trace RAM
Trigger
Sequence

2. LOGIC POD COLOR CODES (Standard) (See Figure 3)

To display a list of the color codes for each probe of the two probe sets on the 32-Channel Logic Pod, depress the S. FUNC key, then 2 on the keypad. See also Section II.A.

3. RS-232C PORT CONFIGURATION
4. ROM EMULATOR TRANSFERS
5. ROM EMULATOR CONFIGURATION
6. ROM EMULATOR EDIT

> ROM Emulator Module-- See Section VIII.B

7. SETUP STORAGE
8. B MEMORY STORAGE
9. COMPARE
10. SEARCH
11. EDIT B MEMORY

> Setup Memory and Setup/Data Memory-- See Section IX.B

IV. OPERATION

(after connection and setup, per Sections II and III)

A. GENERAL

1. RECORDING DATA

To begin recording, depress START. (Recording will automatically stop according to the parameters and conditions selected via the Trigger, Sequence, and Delay setup screens.) To manually stop recording, depress STOP.

Recording begins upon depressing START; recording stops either when the trigger condition has been satisfied and the delay time has expired, or when the STOP key has been depressed. During autosequencing or in the continuous compare mode with the Setup/Data Memory Option (AB-032), recording may also begin automatically after the end of a previous operation.

If the ML4100 user has specified data qualification, but no qualified data has occurred or no valid clocks have occurred, then the ML4100 may be "started" without actually recording any data, and a "No Data Recorded" message will be displayed when "recording" is stopped manually (via the STOP key).

2. DISPLAYING RECORDED DATA

There are three modes for displaying recorded data-- Timing, State, and Disassembly--and they are accessed via the TIMING, STATE, and DISASM keys, respectively, grouped under the "DISPLAY" label on the front panel of the ML4100.

When recording stops, the data will be displayed according to the last display mode selected. (Default at powerup is the Timing display mode.)

To change the display mode at any time during operation, simply depress TIMING, STATE, or DISASM; this will not require the capture of new data. Any changes in the format of these displays via the Format screen will not disturb the recorded data, providing convenient display of the same data in a different number base or bit grouping.

For further details, see Sections IV.B. Timing Display Mode; IV.C, State Display Mode; and IV. D, Disassembly Display Mode.

3. PRINTING DISPLAYED DATA

For ML4100B logic analyzers which have an RS-232C Communication Control Card installed and which are connected via that interface to a printer, the State and Disassembly data displays may be printed as hard-copy reports. The RS-232C interface is contained in the ROM Emulator Module (RE-016), and details of setup when using this interface are given in Section VIII.B.2.

To print a State or Disassembly data display, depress the f4 key while viewing either display. There is no prompt on the screen for printing via the f4 key.

4. MOVING THE CURSOR AND SCROLLING

In the State and Disassembly display modes, the vertical cursor position is fixed at the top data line of the screen, and the up and down arrow keys will scroll the data lines up and down through this fixed position. To instantly position the trigger event at the cursor position (on the top data line), depress either the STATE key or the DISASM key twice while viewing that display.

The top data line of the State and Disassembly display screens corresponds to the vertical cursor position on the Timing display screen.

In the Timing display mode, the cursor may be moved horizontally across any timing line via the left and right arrow keys. The location of the trigger condition occurrence is indicated by a vertical dashed line. To find the exact location of the trigger condition occurrence when the dashed line is shown outside of (to the far left or far right of) a timing line, move the cursor towards the dashed line until the line has scrolled onto the timing line. (Do not confuse the dashed line with the solid vertical line which serves as the cursor.) To instantly position the trigger event at the cursor position, depress the TIMING key twice while viewing the display.

Note that scrolling in any display mode affects the data displayed at the cursor position in the other display modes. In all three displays, the cursor is positioned at the same word in the recorded data.

To change the speed of the cursor when scrolling, depress the f2 key, which toggles the speed back and forth between "Fast" (one screen at a time) and "Slow" (one data word at a time). The current cursor speed mode is displayed at the lower right of each of the three display screens.

To find the current cursor position relative to the trigger condition, depress f1, which erases the bottom two lines of data, displays instead the current cursor position, and allows changing the position.

To change the cursor position (after depressing f1 as above), enter the new position (as the number of samples after or before the trigger condition) via the keypad in the field labeled "Position the cursor _____ clocks", then specify either "A" (After) or "B" (Before) in the other reverse-video field.

To move the cursor to the new position just entered, depress f1 again. If, while viewing a display screen, f1 is depressed and no repositioning of the cursor is desired, simply depress f1 again to return the screen to its former display.

B. TIMING DISPLAY MODE (See Figure 14)

(Be sure to first read the general remarks above, Section IV.A.)

Data is represented by timing lines (diagrams), as in an oscilloscope display; the exact format may be selected via the Display Format screen (Figure 11). Data is displayed (from any active probe) for 12 channels at a time, as selected via the Display Format screen. The 12 channels can be placed in any order, and/or repeated (displayed more than once on the screen), and lines may be left blank (displayed as Probe # "XX").

To scroll through the timing lines horizontally, use the right and left arrow keys.

To magnify (horizontally) the area around the cursor, depress the f3 key; the degree of magnification is a factor of ten, and this is shown (as "X10") on the lower right of the screen in the field labeled "Scale (f3)". Depressing the f3 key again will toggle the magnification off, and the Scale field will display "X1" to so indicate.

A dashed vertical line indicates the trigger condition occurrence; if this line appears outside of the timing line (either to its right or to its left), then the exact location of the trigger condition is not shown, but may be scrolled to by moving the cursor toward it.

When using an internal clock, the ML4100 automatically measures the time represented by the distance (on the timing line) between the trigger condition occurrence and the cursor location. This time is constantly displayed (in nsec, usec, or msec) on the upper right of the screen in the field labeled "Trigger to Cursor".

When using an external clock, distances are measured in units of "clocks". The time (in seconds if an internal clock, or in "clocks" if an external clock) between any two arbitrary locations may be measured by setting a reference line with the cursor at one location, then moving the cursor to the second location.

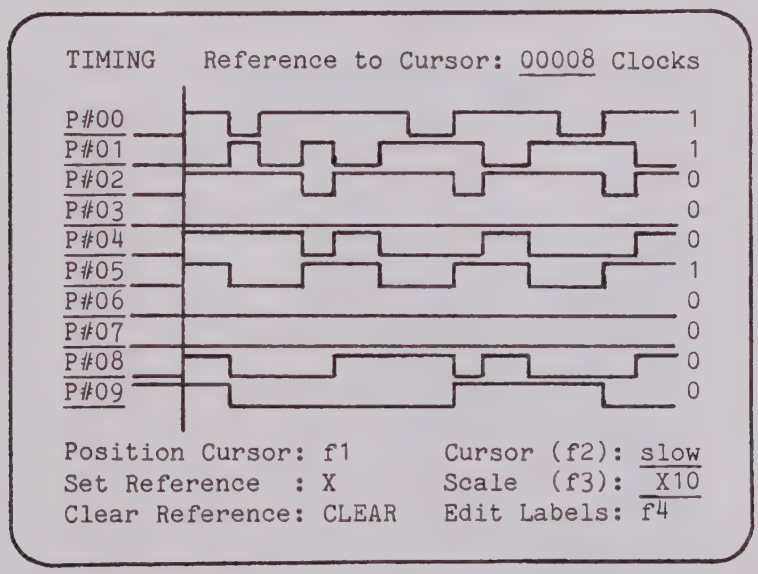
To set a reference line for measuring time, move the cursor to the desired position on a timing line and depress "X" on the keypad. The area between the reference line and the cursor location will be highlighted on the screen. The time represented by the highlighted area is then automatically measured, and will be displayed on the upper right of the screen in the field is "Reference to Cursor".

To clear the reference line and return to the "Trigger to Cursor" field, depress CLEAR on the keypad. To reposition the cursor to the trigger word. depress the TIMING key.

To relabel the timing lines, first depress the f4 key, which erases the bottom three lines of the screen, and displays instead three prompt lines about relabeling. To then advance a label character to the next higher character (e.g., from "C" to "D", or "5" to "6"), depress the f1 key; repeat this as many times as is needed to reach the desired character. To then retreat to the next lower character (e.g., from "C" to "B", or from "5" to "4"), depress the f2 key, and repeat as needed. (These keys automatically repeat when held down.) Use the arrow keys as needed to position the blinking cursor at the character to be changed.

To clear the three relabeling prompt lines from the bottom of the screen, depress the f4 key, which toggles back and forth the prompt lines and the relabeling activity.

Figure 14
TIMING DISPLAY SCREEN



C. STATE DISPLAY MODE (See Figure 15)

(Be sure to first read the general remarks in Section IV.A).

The values in the "STATE" column show sample numbers relative to the trigger. Zero is the trigger condition and is replaced by "TRIG"; negative values are before the trigger, and positive values are after the trigger. Each line of this screen represents one data sample, displayed in bit groupings and number base as selected via the Display Format screen. The trigger condition (or the first data-qualified word in storage which follows the trigger condition) is labeled "TRIG".

To display the trigger condition, scroll up or down via the arrow keys, or move directly to the trigger condition via the STATE key. To cursor to the trigger condition, depress STATE a second time.

To scroll up or down through the data samples, use the arrow keys. (Scrolling up accesses earlier samples, and scrolling down accesses more recent samples.)

To print a hard-copy report of the State display screen (on an ML4100B with an RS-232C installed), depress the f4 key; there is no on-screen prompt for this print function. (The ROM Emulator Module, RE-016, includes an RS-232C.)

Figure 15
STATE DISPLAY SCREEN

```
STATE  BIN  STS  ADR DAT
-0004 0000 0010 003E  00
-0003 0000 0011 009C  21
-0002 0000 0010 009D  00
-0001 0000 0010 009E  08
 TRIG 0000 0011 009F  35
00001 0000 0010 0800  16
00002 0000 0001 0800  15
00003 0000 0011 00A0  23
00004 0000 0011 00A1  35
00005 0000 0010 0801  07
00006 0000 0001 0801  06
00007 0000 0011 00A2  23
00008 0000 0011 00A3  35
00009 0000 0010 0802  07
Position Cursor: f1      Cursor (f2): slow
```

D. DISASSEMBLY DISPLAY MODE (OPTIONAL) (See Figure 16)

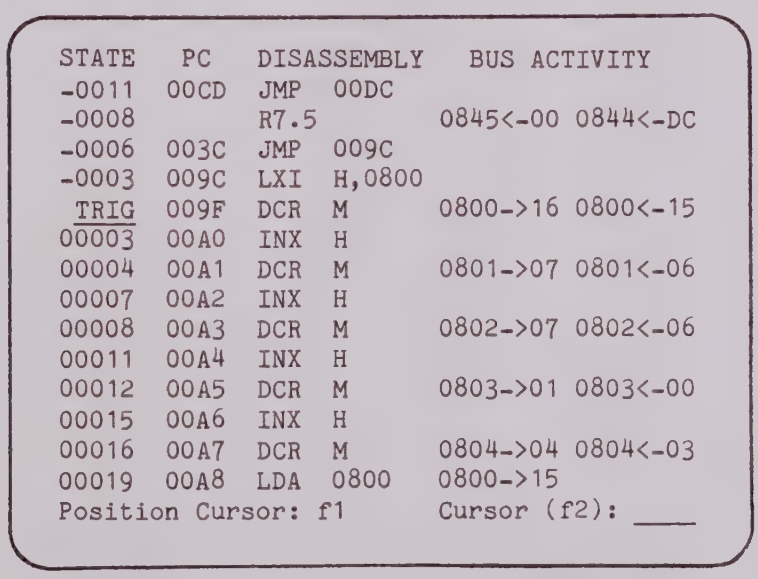
(First read Section IV.A above.)

The Disassembly mode displays data collected via an (optional) microprocessor pod, which enables the ML4100 to display the recorded data as assembly language microprocessor instructions ("reverse assembly").

When data qualification is being used, some ambiguities in the recorded data may result and the disassembler mode may not produce any disassembly output. In this case, the logic analyzer will try to disassemble the recorded data for about 30 seconds, and then will display only the top and bottom lines of the disassembly screen (i.e., will display no data, but just the label and prompt lines).

To print a hardcopy report of the Disassembly display screen (on an ML4100B with an RS-232C installed), depress the f4 key; there is no on-screen prompt for this print function. (The ROM Emulator Module, RE-016, includes an RS-232C.)

Figure 16
DISASSEMBLY DISPLAY SCREEN



(Blank)

V. FOUR-CHANNEL LOGIC POD (LP-040)

A. SPECIFICATIONS

Input Impedance:	50k in parallel with 6 pf
Input Voltage:	+/- 7.9 v
Setup:	5 nsec
Hold:	0.5 nsec
Minimum Detectable Glitch:	40 mV over threshhold for 5 nsec
Maximum Clock Frequency:	100 MHz internal, 75 MHz external
Overdrive:	40 mv

B. OPERATION

WARNING: THE 4-CHANNEL LOGIC POD (LP-040) CONTAINS SEVERAL HIGH-SPEED INTEGRATED CIRCUITS, WHICH DISSIPATE A LARGE AMOUNT OF HEAT. DO NOT USE IT IN ANY POSITION WHICH OBSTRUCTS THE NORMAL AIRFLOW AROUND IT.

1. PROBES

The 4-Channel (Glitch-Capture) Logic Pod has six probes: four for data, one for an external clock, and one for clock qualification. (Clock qualification functions on this pod the same as it does on the 32-Channel Logic Pod, LP-320.)

Each probe consists of two microhooks attached to a length of twisted-pair cable. The all-black hook in each pair is a ground probe, and should be connected to a circuit ground local to the signal being observed. The remaining six hooks are color coded as follows:

FUNCTION:	BODY: (tip)	PLUNGER: (cap)
External Clock	Red	Red
Clock Qualifier	Blue	Blue
Channel 0	Grey	Black
Channel 1	Grey	Brown
Channel 2	Grey	Red
Channel 3	Grey	Orange

2. MEASURING THRESHOLD VOLTAGE

Provision is made for measuring threshold voltage by a pair of wires on the probe set which terminates in a Molex connector. The threshold voltage may be measured here with the contact on the flat end of the connector being at ground.

To adjust the threshold voltage, use the switch and potentiometer slot in the end of the pod. When the switch is towards the screwdriver slot, the potentiometer may be used to vary the threshold over a +/- 7.9-volt range. When the switch is away from the screwdriver slot, the threshhold is fixed at 1.4 v for use in TTL-level circuits.

3. SETUP

Selection of trigger and clock parameters is made in the same manner as for the 32-Channel Logic Pod (LP-320); (see Sections III.C and III.E.).

For the 4-Channel Logic Pod (LP-040), the user may also select either the Glitch or No-Glitch mode via the Format screen. In the No-Glitch mode, data is captured and presented as it is when using the 32-Channel Logic Pod. In the Glitch mode, if an even number of data transitions occur between successive sample clocks, then the displayed data will alternate during the two clock periods following the one in which the glitch occurred. I.e., in the clock period immediately following the glitch, the state will be forced in the direction indicated by the glitch. In the next clock period, the state will be forced to the opposite state, thus insuring that a glitch occurrence (even near a transition) is always displayed as an "isolated bump." Suspected glitches may be examined by switching back and forth between the Glitch and the No-Glitch modes.

(Blank)

VI. 8-BIT MICROPROCESSOR PODS

A. GENERAL

(See also Section II.B, Connection via Microprocessor Pod, and Section IV.D, Disassembly Display Mode.)

Although all 8-bit microprocessor pods operate similarly, there are a few differences among them.

For all 8-bit pods, Data Inputs 0-7 are Data Bus Bits 0-7 (demultiplexed if required). Data Inputs 8-23 are Address, although some bits (high-order) may not be used. Data Inputs 24-31 contain Status and extra Data Input lines. (Data Inputs are labeled on the Format screen as probe numbers.)

On most microprocessor pods, two or four of the high-order inputs are supplied as separate TTL inputs via a connector on the side of the pod. To use these TTL input lines, remove the 8-channel probe set from the 32-Channel Logic Pod (the 20-pin connector) and attach it to the connector on the side of the microprocessor pod.

Clock signals are derived as required from the signals on each microprocessor. (Clock qualification is not used with microprocessor pods.)

The ML4100 will automatically configure itself at powerup for synchronous microprocessor cycle data collection. This configuration may, however, be changed, just as configurations may be changed with other pods.

The default trigger configuration has four fields: "BIN", a binary field for the extra probes; "STS" (Status); "ADR" (Address); and "DAT" (Data). These fields are modifiable as desired. On the bottom of the Trigger screen are the status codes for easy reference.

Details of the data configuration for each microprocessor pod follow.

B. 8I-080 MICROPROCESSOR POD

(Supports Intel 8085, 8035/8039/8040, and 8031/8032 microprocessors)

Data	Data Lines 0- 7	
Address	Data Lines 8-23	(8-19 for 8035/8039/8040)
Status	Data Lines 24-27	
Spare	Data Lines 28-31	(Wires 0-3 on Standard Probe Set of 32-Channel Logic Pod)

STATUS CODES

	8085	8035/39/40	8031/32
Fetch (or PSEN)	0011	X011	X011
Memory Read	0010	X010	X010
Memory Write	0001	X001	X001
I/O Read	0110	X1XX	N.A.
I/O Write	0101	X1XX	N.A.
Interrupt Acknowldgment	0111	N.A.	N.A.
Hold Acknowledgment	1XXX	N.A.	N.A.
Bit 24 =	S0	WR or PSEN	WR or PSEN
Bit 25 =	S1	RD or PSEN	RD or PSEN
Bit 26 =	IO/$\overline{M}$	PROG	N.A.
Bit 27 =	HLDA	N.A.	N.A.

C. 8M-080 MICROPROCESSOR POD

(Supports Motorola 6800, 6802, and 6808 microprocessors)

Data	Data Lines 0- 7
Address	Data Lines 8-23
Status	Data Lines 24-27
Spare	Data Lines 28-31

STATUS CODES

Read	01X1
Write	01X0
Halt/bus avail.	1X0X
Bus available	1XXX
Bit 24 =	R/$\overline{W}$
Bit 25 =	$\overline{HALT}$
Bit 26 =	VMA
Bit 27 =	BA

SWITCH SETTINGS

For proper operation of the 6800/6802/6808 microprocessor pod, follow the switch settings. The switches are located on the same side of the pod as the auxiliary probe connector, and are labeled 1, 2, 3, and 4. Up (toward the pod label) is open; down is closed.

	6802 and 6808	6800
Switch 1	Open	Write Clocking: Open--Fall of 02 Closed--fall of DBE
Switch 2	(Unused--leave open)	(Unused--leave open)
Switch 3	(Unused--leave open)	(Unused--leave open)
Switch 4	(Unused--leave open)	(Unused--leave open)

NOTE: To detect a reversed pod connection error, the 6800 pod loads the target processor's VCC pin with 200 ohms to VSS. If this 25-mA load is unacceptable, remove R7 from the pod's circuit board, or replace R7 with a resistor of a higher value.

D. 8M-089 MICROPROCESSOR POD

(Supports Motorola 6809 and 6809E microprocessors)

	Data Lines	
	6809	6809E
Data	0- 7	0- 7
Address	8-23	8-23
Status	24-27	24-29
Spare	28-31	30-31

STATUS CODES

Last Instruction Cycle	N.A.	X1XXXX
Read	00X1	XX00X1
Write	00X0	XX00X0
Interrupt Acknowledgment	01XX	XX01XX
Sync. Acknowledgment	10XX	XX10XX
Bus Grant Due to Halt	110X	XX110X
Bus Grant	11XX	XX11XX
Bit 24 =	R/$\overline{W}$	R/$\overline{W}$
Bit 25 =	$\overline{HALT}$	$\overline{HALT}$
Bit 26 =	BS	BS
Bit 27 =	BA	BA
Bit 28 =	(Side Probe 0)	LIC
Bit 29 =	(Side Probe 1)	AVMA
Bit 30 =	(Side Probe 2)	(Side Probe 0)
Bit 31 =	(Side Probe 3)	(Side Probe 1)

NOTE: To detect a reversed pod connection error, the 6809 pod loads the target processor's VCC pin with 200 ohms to VSS. If this 25-mA load is unacceptable, remove R7 from the pod's circuit board, or replace R7 with a resistor of a higher value.

E. 8R-065 MICROPROCESSOR POD

(Supports Rockwell 6502, 6512, 65C02, 65C102, and 65C112 microprocessors)

Data	Data Lines 0- 7
Address	Data Lines 8-23
Status	Data Lines 24-27
Spare	Data Lines 28-31

STATUS CODES

	6502 & 6512	65C02, 65C102, & 65C112
Fetch	1X11	1X11
Memory Read	0X11	0X11
Memory Write	0XX0	0X10
Set Overflow Bit	X0XX	X0XX
Bit 24 =	R/$\overline{W}$	R/$\overline{W}$
Bit 25 =	RDY	RDY
Bit 26 =	S.O.	S.O.
Bit 27 =	SYNC	SYNC

SWITCH SETTINGS

For proper operation of the 6502 microprocessor pod, follow the switch settings below. The switches are located on the same side of the pod as is the auxiliary probe connector. and are labeled 1, 2, 3, and 4. Up (towards the pod label) is open; down is closed.

	6502	65C02, 65C102, 65C112	6512
Switch 1	Open	Open	Write Clocking: Open--Use S2 choice Closed--Fall of DBE
Switch 2	Open	Open	Read Clocking: Open--Fall of Φ2 Closed--Rise of Φ1
Switch 3	Closed	Open	Open
Switch 4	Open	Open	Closed

NOTE: To detect a reversed pod connection error, the 6502 pod loads the target processor's VCC pin with 220 ohms to VSS. If this 23 mA load is unacceptable, remove R7 from the pod's circuit board, or replace R7 with a resistor of a higher value.

F. 8Z-080 MICROPROCESSOR POD

(Supports Zilog Z80 microprocessor)

Data	Data Lines 0- 7
Address	Data Lines 8-23
Status	Data Lines 24-27
Spare	Data Lines 28-31

STATUS CODES

Fetch	1011
Memory Read	1010
Memory Write	1001
I/O Read	1110
I/O Write	1101
Interrupt Acknowledgment	1111
Bus acknowledgment	0XXX
Refresh	1000

Bit 24 =	Write or Fetch or Interrupt Acknowledgment
Bit 25 =	Read or Fetch or Interrupt Acknowledgment
Bit 26 =	I/O or Interrupt Acknowledgment
Bit 27 =	$\overline{\text{BUSAK}}$

G. 8N-080 MICROPROCESSOR POD

(Supports National Semiconductor NSC800 microprocessor)

Data	Data Lines 0- 7
Address	Data Lines 8-23
Status	Data Lines 24-28
Side Probes (3)	Data Lines 29-31

STATUS CODES

Fetch	X1011
Memory Read	X1010
Memory Write	X1001
I/O Read	X1110
I/O Write	X1101
Interrupt Acknowledgment	X1111 (Pod's encoding, not microprocessor's)
Bus Acknowledgment	X0XXX
Refresh	X1100 (Pod's encoding, not microprocessor's)
Halt	X1000
Processor was reset	1XXXX

Bit 24 = S0, High for Write or Fetch or Interrupt Acknowledgment

Bit 25 = S1, High for Read or Fetch or Interrupt Acknowledgment

Bit 26 = IO/$\overline{M}$, High for I/O or Interrupt Acknowledgment

Bit 27 = $\overline{BUSAK}$

Bit 28 = Reset: High for each processor cycle during which Reset Out was high. plus one additional cycle

Bit 29 = Side Probe 0

Bit 30 = Side Probe 1

Bit 31 = Side Probe 2

During operation with internal clock (i.e., asynchronous to the microprocessor), only the data lines are totally "transparent". Both address bytes are registered with D-type registers, and changes only on the falling edge of ALE. The Status lines, while not registered, are affected by the encoding logic which marks Refresh cycles (if in NSCREF mode), and Interrupt Acknowledgment cycles.

VII. STD BUS POD (8P-080)

A. INTRODUCTION

The STD Bus Pod enables the ML4100 to capture bus cycles asynchronously (with the internal clock) or synchronously (with the external clock). To install the STD Bus Pod, insert it into an unoccupied bus slot or use it as an extender card, by plugging the card which normally occupies the bus slot into the STD Bus Pod (at J1), then inserting the STD Bus Pod into the bus slot.

B. POD MODES

The pod has three modes, as shown on the Format setup screen: STDBUS, STDINT, and STD24B. The first 24 data lines are the same in each mode; the remaining 8 data lines differ among the modes, and, by altering jumpers on the board, the signals presented in the first two modes can be selected by the user. Data lines for each of the three modes are shown in the table below.

Data Lines	STDBUS Mode (Default)	STDINT Mode	STD24B Mode
0- 7	Data	Data	Data
8-23	Address (lower 16 bits)	Address (lower 16 bits)	Address (lower 16 bits)
24	RD*	INTRQ* (see jumper options)	- - - - - - - - |
25	WR*	INTAK*	|
26	IORQ*	IORQ*	|
27	MEMRQ*	PCI	|
28	IOEXP (if IORQ* is active) MEMEX (if MEMRQ* is active)	PCO	Address (upper 8 bits) | |
29	BUSRQ*	REFRESH*	|
30	BUSAK*	STATUS0*	|
31	WAITRQ*	STATUS1*	- - - - - - - -

Data lines 24-31 in the STDBUS (default) mode, with default jumpering, present the main data transfer control signals of the bus. These lines in the STDINT mode, with default jumpering, present primarily those signals related to interrupts.

In the STD24B mode, data lines 24-31 present the upper 8 bits of a 24-bit address. These bits are clocked from the data bus into an octal register on the pod at the rising edge of the MCSYNC* bus signal; the outputs of this register drive data lines 24-31. None of the jumpering options affects the STD24B mode. (NOTE: Early versions of the ML4100 code display these bits in two 4-bit binary fields.)

C. JUMPERING OPTIONS

1. DATA LINES 24-31

In the STDBUS and STDINT modes, the signals presented on Data Lines 24-31 may be selected by the user. Any eight signals may be selected from the signals available at jumper posts to be displayed in either of these two modes. The table below shows the signals available, the mode/data-line destinations, and the default connections.

Sources		Destinations		
Signal	Jumper Pin	Jumper Pin	Mode	Data Line
IOEXP/MEMEX	JP3- 1	JP3- 2	STDBUS	28
PCO	JP3- 3	JP3- 4	STDINT	28
BUSAK*	JP3- 5	JP3- 6	STDBUS	30
STATUS0*	JP3- 7	JP3- 8	STDINT	30
BUSRQ*	JP3- 9	JP3-10	STDBUS	29
REFRESH*	JP3-11	JP3-12	STDINT	29
WAITRQ*	JP3-13	JP3-14	STDBUS	31
STATUS1*	JP3-15	JP3-16	STDINT	31
MEMRQ*	JP4- 1	JP4- 2	STDBUS	27
PCI	JP4- 3	JP4- 4	STDINT	27
IORQ*	JP4- 5	JP4- 6	STDBUS	26
IORQ*	JP4- 7	JP4- 8	STDINT	26
RD*	JP4- 9	JP4-10	STDBUS	24
INTRQ*	JP4-11	JP4-12	STDINT	24
WR*	JP4-13	JP4-14	STDBUS	25
INTAK*	JP4-15	JP4-16	STDINT	25
SYSRESET*	JP5- 1			
PBRESET*	JP5- 2			
NMIRQ*	JP5- 3			
CLOCK*	JP6- 1			
CNTRL*	JP6- 2			

2. REFRESH CYCLES

During synchronous recording of bus cycles, the pod will record each Read, each Write, and each Interrupt Acknowledge cycle Also Refresh cycles may be recorded if a card on the bus supplies them. Either Z80 or NSC800 Refresh cycles may be captured. or Refresh cycles may be ignored according to the table below.

Refresh Type	Jumpers Patch	
	(JP1)	(JP2)
Refresh ignored (default)	2-3	(Don't Care, default = 2-3)
Z80	1-2	2-3
NSC800	1-2	1-2

3. INTRQ* AND BUSRQ*

When the pod is used as an extender card, and the card plugged into J1 is one which may issue INTRQ* and/or BUSRQ*, then the corresponding request signal available to the ML4100 may be either the actual request signal on the STD Bus or the request signal emanating from the card. The table below shows the options available.

Request Signal	Connections on JP7
BUSRQ* Source:	
Bus (default)	1-3
Card	3-5
INTRQ* Source:	
Bus (default)	2-4
Card	4-6

Note that, if the card must receive the request from the bus, then the request source must be the bus; the option of having the card as source is not available in that situation.

D. ASYNCHRONOUS OPERATION

When the STD Bus Pod is used asynchronously, the user should note that the signals traveling to the ML4100 typically pass through two buffers (one high-speed CMOS buffer and one LS TTL buffer), plus a resistor network.

Since different signals pass through different buffers, the timing relationships between signals will be altered slightly, due to differences in path delays. This timing skew should be less than 20 ns, with one exception: the address signals A0 through A15 are intentionally delayed approximately 20 ns more than all other signals for proper synchronous operation with the Z80 STD Bus. (The Z80 CPU, after an opcode fetch, very quickly outputs the refresh address, and without the delay the address of the opcode fetch may not be captured properly at the clock edge occurring at the end of the fetch.)

To remove this added delay in the address lines during asynchronous operation, change the resistor networks in Locations 5E and 6C to 270-ohm, individual-resistor networks (16 pins, 8 resistors). These locations are socketed. Save the original resistor networks for synchronous operation.

VIII. ROM EMULATOR MODULE (RE-016)

(Blank)

A. INTRODUCTION

The ROM Emulator Module option enables the ML4100 to emulate one or more 27XX-series EPROMS, giving it a powerful tool for debugging code in microprocesor-based systems. Code can be downloaded to the emulator and then patched as required.

The ROM Emulator Module consists of two assemblies: the ROM Emulator Control Board, Catalog Number RS-232, (an internal card) and the ROM Emulator Pod, RP-016, (an external pod box). With one pod, up to 16 Kbytes of memory may be emulated, and configured as four 2716s, four 2732s, two 2764s, or one 27128. With a second pod, twice the memory can be emulated. (The ML4100 accommodates no more than two ROM Emulator Pods.)

B. SETUP AND EDITING

1. GENERAL

To access the ROM Emulator Module, depress the Special Function key on the ML4100 front panel, then select one of four screens via the keypad, as shown below:

Key	Function
3	RS-232C Port Configuration
4	ROM Emulator Transfers
5	ROM Emulator Configuration
6	ROM Emulator Edit

Before using any ROM Emulator, the user must first specify the configuration; failure to do so will probably cause error messages and/or incomplete function of the ROM Emulator. Before making any data transfers (uploading or downloading), the user must first specify the configuration of the RS-232C port, via the RS-232 Port Configuration screen. Both these screens are accessed by depressing the Special Function key, then by depressing a key on the keypad to select the appropriate screen (as shown on the table above).

To specify setup parameters on any of the four screens named above, cursor to a field, then, using the hexadecimal keypad on the front panel, either select one of the given parameters as indicated by the screen prompting labels, or enter the desired value directly into the field.

2. CONFIGURATION OF RS-232C SERIAL PORT (See Figure 17)

The RS-232C Serial Port Configuration screen is used to configure the serial port before downloading or uploading information to or from the ROM Emulator. It allows the user to specify the following parameters:

a. Baud rate (from 8 given rates);

b. Parity;

c. Number of bits per character;

d. How long the port should delay after transmitting a character;

e. How much it should delay after a carriage return (for devices which are slow to receive characters);

f. Whether it should add linefeeds to carriage returns;

g. Whether it should recognize XON & XOFF; and

h. Whether it should recognize CLEAR TO SEND ("CTS").

The user should specify the above configuration parameters such that they are compatible with whatever device is communicating with the ML4100 via the RS-232C interface.

The baud rate is the rate (in bits per second) at which messages will be sent and received by the ML4100.

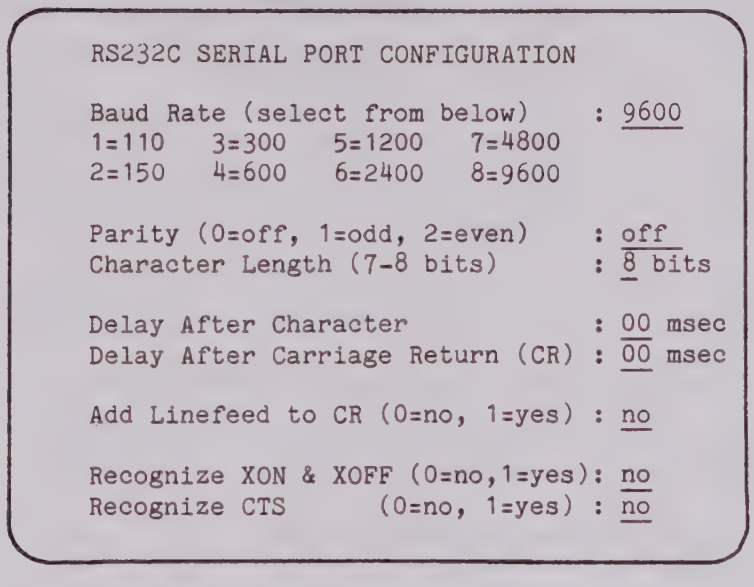
Specification of delays (Items d and e above) is needed when interfacing with devices which cannot accept character transmission rates at full speed at the baud rate selected by the user, so that data will not be lost when transmitting from the ML4100. A delay (of up to 99 msec) may be specified after each character or after each carriage return. However, if a handshaking protocol is implemented when transmitting from an ML4100 to a slower device, then no delay is needed. (The ML4100 can accept data continuously at 9600 baud.)

When using the RS-232C interface to transmit data or screen displays to a CRT or a printer, some systems automatically will add a linefeed after each carriage return; if the system does not, the user should specify adding a linefeed to prevent overprinting each line on the previous line. When communicating with a computer, higher throughput is achieved by not adding a linefeed after each carriage return.

For details of the parameter labeled "Recognize XON & XOFF," see Section C, "Hardware Considerations."

NOTE: On the ML4100B, this same screen is used to configure the RS-232C port for functions other than ROM emulation, such as screen print (accessed via the f4 key when on the State or Disassembly screens) and A/B Memory functions.

Figure 17
RS-232C SERIAL PORT CONFIGURATION


RS232C SERIAL PORT CONFIGURATION
Baud Rate (select from below) : 9600
1=110 3=300 5=1200 7=4800
2=150 4=600 6=2400 8=9600
Parity (0=off, 1=odd, 2=even) : off
Character Length (7-8 bits) : 8 bits
Delay After Character : 00 msec
Delay After Carriage Return (CR) : 00 msec
Add Linefeed to CR (0=no, 1=yes) : no
Recognize XON & XOFF (0=no,1=yes): no
Recognize CTS (0=no, 1=yes) : no

3. DATA TRANSFERS TO AND FROM ROM EMULATOR (See Figure 18)

The ROM Emulator Transfers screen is used to upload or download data to or from the ROM Emulator, and allows the user to specify the following:

a. Which pod to download to or upload from;

b. Lowest address and highest address (to be transferred to or from);

c. Transfer format (Intel hex, Tektronix hex, and Motorola S Records (see Appendix C for Intel hex format);

d. Transfer type, either copy or verify (for verifying data already in ROMs in case previous edits have been forgotten); and

e. Transfer direction (to or from the ML4100 port).

Before downloading to a ROM area, it must be enabled, via the Configuration screen (described below in Section 4).

If two ROM Emulator Pods are attached to the ML4100, the pod closest to the ML4100 is considered the "first" pod.

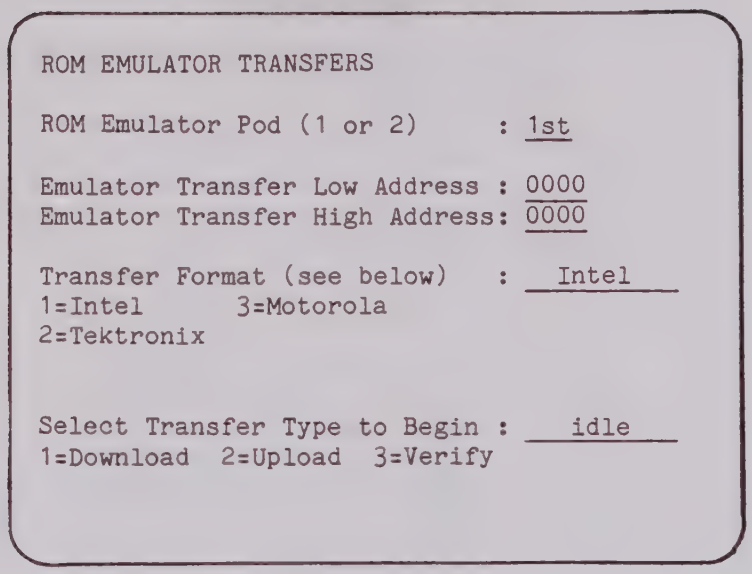
There are two Intel transfer formats and three Motorola transfer formats; only one of each appears on the Transfer screen. In each case, the transfer formats are upward compatible; i.e., Intel has both 8-bit and 16-bit formats. The 8-bit format is a subset of the 16-bit format, and the unit will allow the 16-bit format to be downloaded in any case. During upload it will check the address size; if it is 20 bits, it will upload using Intel 16-bit format, and if it is 16 bits, it will upload using Intel 8-bit format.

Similar decisions are made on the Motorola format, which allows different address lengths to be downloaded (16-, 24-, or 32-bit). The ML4100 will accept any of the three on download, and will check the address width selected via the configuration screen to determine which format to use to upload.

To transfer data to or from the ML4100, cursor to the "Select Transfer Type To Begin" field and depress the 1, 2, or 3 key (to select Download, Upload, or Verify, respectively). (If the user has made no selection, "idle" will be displayed in the field.) When the data transfer has been completed, the low and high address displayed on the screen will automatically be updated to the actual lowest and highest addresses transferred. If this display shows "FFFF" for the low address and "0000" for the high address, then no bytes were transferred.

To stop a data transfer, depress STOP.

Figure 18
ROM EMULATOR TRANSFERS


ROM EMULATOR TRANSFERS
ROM Emulator Pod (1 or 2) : 1st
Emulator Transfer Low Address : 0000
Emulator Transfer High Address: 0000
Transfer Format (see below) : Intel
1=Intel 3=Motorola
2=Tektronix
Select Transfer Type to Begin : idle
1=Download 2=Upload 3=Verify

4. ROM EMULATOR CONFIGURATION (See Figure 19)

The ROM Emulator Configuration screen enables the user to configure the way in which the ROM Emulator appears to the hardware, as well as to the user, when editing or downloading. The user may select:

a. Numeric base to be used in editing (octal or hexadecimal);

b. Address and data widths in bits (address widths may be from 11 to 24 bits, and data width may be 8, 16, or 32 bits);

c. Addressing mode (either one address per byte or one address per word, where a word is the selected data width);

d. Device type for a given ROM pod; and

e. Base addresses for the ROMs being emulated.

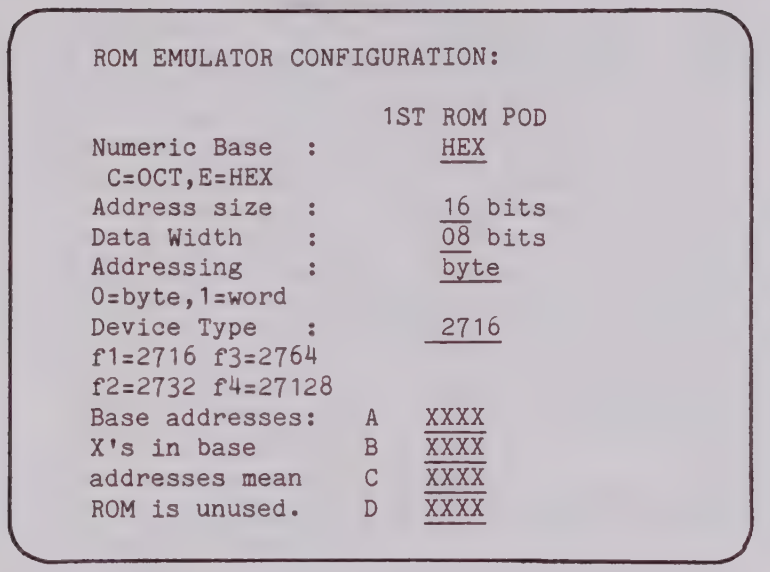
All of the above are selectable for each of the two ROM pods, if two ROM pods are installed. An exception is when the user selects 32-bit data width, in which case both ROM Emulator Pods are necessary to provide 32 bits of data. Also, if the device selected is 27128, then the 32-bit width is not available, and two ROM pods are required to emulate 16 bits. If a nonstandard data width is entered (other than 8, 16 or 32 bits), then it is rounded up to the nearest above value. Data widths over 32 bits are not allowed.

The ROM Emulator Configuration screen base address fields enable the user to set base addresses for the ROMs being emulated, making editing and downloading convenient. For example, when the ROM Emulator is emulating four 2716s, one ROM may have a base address of 0, the next may have a base address of 800 hex, and the next, a base address of 1000 hex, etc.

To specify base addresses for the ROMs being emulated, cursor to each Base Address field on the Configuration screen and use the keypad to enter the address in each one. Some base address must be specified (even if it is 0) to enable areas of ROM; if an area is not enabled, it cannot be downloaded to or edited.

To disable a ROM, enter Xs ("Don't Cares") in the base address fields.

Figure 19
ROM EMULATOR CONFIGURATION



5. EDITING DATA IN THE ROM (See Figure 20)

The ROM Emulator Edit screen enables the user to edit data in the emulated ROM. It has two types of alterable fields: "Address" and "Data."

To access a particular address in ROM, either use the Up and Down arrow keys to scroll the cursor to the particular Address field, or scroll to any Address field and use the keypad to edit the field by entering the address to be accessed. (The new address will be displayed as soon as the cursor exits the field.) While editing, the user can view six locations above and six locations below the field being edited, and so knows the location in ROM, what has been done, and what still needs to be done.

To edit a Data field, cursor to the corresponding Data field and enter the new value via the keypad. (The new data will be displayed and the location written as soon as the cursor exits the field.)

To restore a field being edited to its previous value, depress the f1 key before exiting the field.

To access the other ROM pod in multipod applications, depress the f2 key, which toggles back and forth between "first" and "second" pod.

To cause a checksum to be computed, first enter the low address for the checksum range in an Address field. Next cursor to the corresponding Data field, and back to the Address field, and enter the high address for the checksum range; then depress the f3 key.

To fill memory with desired constant data, first enter the low address of the fill range in an Address field. Next cursor to the corresponding Data field, and enter the fill range data. Then cursor back to the Address field and enter the upper address of the fill range. Finally, depress the f4 key.

Figure 20
ROM EMULATOR EDIT

```
ROM EMULATOR EDIT:    Selected Pod: ____

 ADDRESS DATA         Scroll to, or enter,
 00001   40           address.  Edit data.
 00002   C4
 00003   02           f1 cancels an entry.
 00004   00
 00005   30           f2 toggles ROM pod #.
 00006   15
 00007   D1           Sum:  Enter low addr.
 00008   C3           To data & back.  Key
 00009   20           in high addr, hit f3.
 00010   00
 00011   00           Fill: Enter low addr.
 00012   00           & the fill data.  Key
 00013   00           in high addr, hit f4.
```

C. HARDWARE CONSIDERATIONS

The ROM Emulator hardware is capable of detecting whether a ROM socket has been plugged in backwards, has not been plugged in, or has been plugged into a system with no power. The software will make checks when the user exits the Edit screen. These checks, if they run into an inconsistency, will flag an error. Selecton of ROMS to be used is accomplished on the Configuration screen. If the base address is set with "Xs", it is treated as unused.

If a parity error occurs during transfers, a beep will be generated. If an overrun or framing error occurs, or if an illegal character is encountered, then the transfer will immediately stop. Transfers will also be stopped by depressing the STOP key. On download, a byte is entered into the ROM Emulator if the address of that byte is between the lower and upper address bounds entered on the Transfers screen and is also being emulated, as defined by the blocks of one of the address bases determined by the length of the device being emulated (either 2 Kbytes up to 16 Kbytes) and by the base address of the given ROMs.

NOTE: While editing or downloading or uploading from the ROM Emulator, it is not accessible as ROMs. Therefore, it will probably be necessary to reset the target after any of these operations.

There are two handshake methods of starting and stopping the RS-232C data stream to or from the ML4100. One method uses the RTS and CTS lines of RS-232C; the other uses DC1 and DC3 (XON and XOFF) of U. S. ASCII. The selection of these is described in Section B.2, above.

The following pins are on the back panel Data Interface connector:

Pin No.	Signal Name (per RS-323C)	Input to ML4100	Output from ML4100
1	Safety ground	X	X
2	TXD		X
3	RXD	X	
4	RTS		X
5	CTS	X	
7	Signal ground	X	X

The ML4100 is a "DTE" device, i.e., it appears at the RS-232C interface as would a standard CRT terminal.

Note that ROM Emulator control cards with serial numbers of 001 through 025 do not pull Pin 4 high.

D. ERROR CODES

Error Code	Meaning
1	Checksum error
2	Nonhex character
3	Invalid character
4	Verify fail
16	Overrun error*
32	Framing error*
64	Break detect*

* Combinations of these three errors may be shown as the sum of their error codes.

IX. SETUP MEMORY OPTION (SM-080) AND SETUP/DATA MEMORY OPTION (AB-032)

(blank)

A. INTRODUCTION

The ML4100 has two memory options: the Setup Memory Option (SM-080) and the Setup/Data Memory Option (AB032). The Setup/Data Memory Option has all the basic capabilities of the Setup Memory Option, plus enhanced data memory capabilities as well. With the addition of either option, the ML4100 logic analyzer becomes an even more effective instrument for data analysis, portable field service work, and production test.

1. SETUP MEMORY OPTION (SM-080)

The Setup Memory Option saves eight complete machine setups (trigger words, timing line labels, format specifications, etc.) in nonvolatile memory. These setups are retained when power is turned off.

This storage of setup files permits rapid, convenient paging between different analysis methods for any given piece of captured data (i.e., paging between different sets of timing lines with alternate labels and positions on the display screen.) Similarly, without re-entering any labeling or triggering conditions, the user can switch from a 32-channel, state, data-qualified mode to a 4-channel, 100-MHz mode immediately, and have all of the data capture parameters already defined. Also, complex setup data need not be lost when there are interruptions and the user needs to be away from the logic analyzer for long periods of time.

2. SETUP/DATA MEMORY OPTION (AB-032)

The Setup/Data Memory option, in addition to saving eight complete machine setups, expands the ML4100's capabilities further by adding a B (reference) Memory. The B Memory is the same size as the machine's A (record) Memory (1,000 samples deep at 32 channels wide, and 8,000 samples deep at 4 channels wide). With the B Memory in this option, Compare and Search functions have been added to the ML4100 to facilitate more extensive data analysis. In addition, B Memory may be edited to allow the Search and Compare functions to be more selective. B Memory may also be saved in nonvolatile memory.

3. INSTALLATION

Both the SM-080 and AB-032 options are contained on a single option card, which may be field-installed by the user.

B. OVERALL OPERATION

The Setup Memory function of storing setups is operated via the Setup Storage screen. The Setup/Data Memory Option uses the Setup Storage screen, plus four other screens: B Memory Storage, Compare, Search, and Edit B Memory.

To access the Setup Memory functions, first depress the Special Function key, then depress the key on the keypad which corresponds to the function:

Key	Function
7	Setup Storage
8	B Memory Storage*
9	Compare*
A	Search*
B	Edit B Memory*

* In AB-032 only

1. SETUP STORAGE (See Figures 21a, 21b, and 21c)

The Setup Storage screen is shown in three figures in this section. Figure 21a shows the screen when the user has positioned the cursor in the Action field; Figure 21b shows it with the cursor in the Name field; and Figure 21c shows it with the cursor in the Status field. User activities via these three screens are detailed in Sections a, b, and c below.

On this screen, the Mode field indicates the setup file mode, i.e., 8085, Z80, LOGIC4, etc. The correct pod must be connected to the ML4100 before recalling a setup file or a Pod Conflict error will occur.

In the Setup column, an asterisk to the left of the file number indicates the current setup file in use (last file accessed). An "M" to the right of the file number indicates that the setups currently in the ML4100 have been modified and differ from the values in the file.

a. Saving Setups (See Figure 21a)

Up to eight ML4100 setup files can be saved. A setup file contains all setup parameters selected by the user (including those of the RS-232 Communications Control Card and the RE-016 ROM Emulator), except for search words.

To save a setup file, cursor to its action field and depress the f1 key (SAVE setups), then depress f1 again (to verify the action). To recall a setup file, cursor to its action field and depress the f2 key (RECALL setups), then depress f2 again (to verify the action); current setups will be replaced by the setups recalled from the file selected.

In the Setup column, an asterisk to the left of the file number indicates the current setup file in use (last file accessed). An "M" to the right of the file number indicates that the setups currently in the ML4100 have been modified and differ from the values in the file.

If the RS-232 Communication Control Card or the RE-016 Rom Emulator Module has been installed in the ML4100, then machine setups may also be uploaded and downloaded via this screen. To upload or download setups, cursor to an Action field and depress the f3 key ("UPLOAD B") or the f4 key ("DOWNLOAD B"). Setups are uploaded to or downloaded from machine memory rather than the files themselves, so the location of the cursor in the action column is unimportant. Downloaded setups may then be saved in a setup file.

Figure 21a
SETUP STORAGE SCREEN
(Cursor in Action field)

SETUP FILE DIRECTORY

SETUP	NAME	STATUS	MODE	ACTION
* 1	TEST1	UNPROTECTED	Z80	SAVE
2	TEST2	UNPROTECTED	8085	----
3	____	AVAILABLE		----
4	PROD1	UNPROTECTED	8085	RECALL
5	PROD2	UNPROTECTED	8085	RECALL
6	PROD3	UNPROTECTED	8085	RECALL
7	PROD4	UNPROTECTED	8085	RECALL
8	____	UNPROTECTED	LOGIC4	------

CURSOR TO ACTION FIELD AND USE f KEYS
f1=SAVE setups f2=RECALL setups
f3=UPLOAD setups f4=DOWNLOAD setups
POWERUP FIELD: 1
AUTOSEQUENCING: start file: 4 end file: 7
*=setup file in use M=setups modified

b. File Names (See Figure 21b)

Setup files can be assigned names of up to six characters, before saving the file. To enter a setup file name, cursor to each space in the Name field and enter one character at a time, using the f1 and f2 keys to alphabetically advance or retreat characters, until the desired character is displayed in each space. The f3 key (clear field) may be used to clear the entire name field.

To change a setup file name, edit the name field one character at a time, using the f1 and f2 keys, then save the file again. If a file name is changed without the file being resaved, a warning message will be issued if an attempt is made to exit the screen.

Figure 21b
SETUP STORAGE SCREEN
(Cursor in Name field)

```
                SETUP FILE DIRECTORY
SETUP    NAME      STATUS       MODE    ACTION
* 1      TEST1   UNPROTECTED     Z80     SAVE
  2      TEST2   UNPROTECTED    8085     ----
  3              AVAILABLE               ----
  4      PROD1     PROTECTED     8085    RECALL
  5      PROD2     PROTECTED     8085    RECALL
  6      PROD3     PROTECTED     8085    RECALL
  7      PROD4     PROTECTED     8085    RECALL
  8              UNPROTECTED   LOGIC4    ----
CURSOR TO NAME FIELD AND USE f KEYS
f1=advance character  f2=retreat character
f3=clear name
POWERUP FIELD: 1
AUTOSEQUENCING: start file: 4 end file: 7
*=setup file in use      M=setups modified
```

c. Status (Protected, Unprotected, Available) (See Figure 21c)

Setup files normally have a status of "Unprotected," but may also be "Protected." To protect a setup file, cursor to the Status field, depress the f3 key (protected), then save the file. When the file is saved, a security code will be requested; this is a 2-digit hexadecimal number which is entered on the keypad. (If more than two keys are depressed, only the last two are used.) The security code will not be displayed on the screen. This security code will be required to write over the file in the future, and it is strongly recommended that it be written down and stored for future use.

To change a setup file from Protected to Unprotected, cursor to the Status field, depress the f2 key, and save the file, using the correct security code. To erase a setup file, cursor to the Status field, depress the f1 key to change its status to Available, and save the file.

Figure 21c
SETUP STORAGE SCREEN
(Cursor in Status field)

```
                SETUP FILE DIRECTORY
SETUP    NAME      STATUS       MODE    ACTION
* 1      TEST1   UNPROTECTED     Z80     SAVE
  2      TEST2   UNPROTECTED    8085     ----
  3              AVAILABLE               ----
  4      PROD1    PROTECTED     8085     RECALL
  5      PROD2    PROTECTED     8085     RECALL
  6      PROD3    PROTECTED     8085     RECALL
  7      PROD4    PROTECTED     8085     RECALL
  8              UNPROTECTED   LOGIC4
CURSOR TO ACTION FIELD AND USE f KEYS
f1 = available          f2 = unprotected
f3 = protected
POWERUP FIELD: 1
AUTOSEQUENCING: start file: 4 end file: 7
*=setup file in use      M=setups modified
```

d. Powerup Field

To designate a powerup setup file, cursor to the Powerup field and enter the setup file number; this setup file will then be automatically recalled upon powerup. To disable the powerup setup, cursor to the Powerup field and enter 0.

e. Autosequencing Start and End (AB-032 only)

The autosequencing start and end file fields apply only to the AB-032 option. These fields select the setup files to be used in autosequencing, a method of automatically running up to eight different tests a preset number of times. (See Section IX.3.c, Autosequencing, for operating details.)

2. B MEMORY STORAGE (AB-032 only) (See Figures 22a, 22b, and 22c)

Data captured by the ML4100 can be copied to B Memory to be used as reference data for Compare operations. If nonvolatile storage is wanted, this B Memory can be saved in a B Memory file.

The user has the option of configuring B Memory storage into one full-sized 4000-byte segment, two 2000-byte segments, four 1000- byte segments, or eight 500-byte segments. This feature permits storage of up to eight reference memories if the user decides (as in many cases) that only a portion of the trace buffer needs to be used for a Compare. Note the distinction between B Memory and a B Memory file; the machine contains only one B Memory, but may have up to eight saved B Memory files.

The B Memory Storage screen is shown in three figures in this section. Figure 22a shows the screen when the user has positioned the cursor in the Action field; Figure 22b shows it with the cursor in the Name field; and Figure 22c shows it with the cursor in the Status field. User activities via these three screens are detailed in Sections 2a-e below.

The width of the stored data (32, 16, 8, or 4 channels) is shown for each file. The selected word width of the ML4100 must match the width of the B memory file or an error will occur on recall.

As with the Setup Storage screen, an asterisk to the left of the file number indicates the current B Memory file in use (last file accessed). An "M" to the right of the file number indicates that B Memory currently in the ML4100 differs from the file indicated by the asterisk.

a. B Memory Configuration (See Figure 22a)

To select the configuration for saved B Memory, cursor to the B Memory Configuration field and enter 1, 2, 4 or 8 to specify the number of memories. Changing the configuration will change the size of the B Memory segments. As soon as B Memory is saved in a B Memory file, the B Memory directory is automatically updated to show the new number of files available.

For example, Figure 22a shows that there are eight B memory files available for use; changing the configuration to one would set the B Memory size to a 4000-byte memory. The B Memory Storage screen, however, would still show the eight files saved in nonvolatile memory, until a Save action is executed. At that time, the B Memory Storage screen would be updated to show only one B Memory file.

b. Copying Data (See Figure 22a)

To copy captured data in A Memory to B Memory, depress the "X" key. If saved B Memory is configured to any size except one, start and end addresses for the copy will be requested. If B Memory is configured to one, the entire A memory will be copied. If there is insufficient data in A Memory to fill B Memory (a trigger occurred before a full buffer of data was recorded), then B Memory will be padded with Don't Cares. B Memory may now be edited or used for Compare and Search operations.

c. Storing Data in Saved B Memory (See Figure 22a)

There must be data in B Memory before executing a Save; this data can be generated either by executing a copy A to B, a recall of a B Memory file, or a download of a B Memory.

To save B memory in nonvolatile storage, cursor to the Action field and depress the f1 key ("SAVE B"), then depress f1 again to verify the action. To recall a B Memory file, cursor to the Action field and depress the f2 key ("RECALL B"), then depress f2 again to verify the action. The current B Memory will be overwritten by the B Memory file selected.

As with the Setup Storage screen, an asterisk to the left of the file number indicates the current B Memory file in use (last file accessed). An "M" to the right of the file number indicates that B Memory currently in the ML4100 differs from the file indicated by the asterisk.

If the RS-232 Communication Control Card or the RE-016 ROM Emulator Module has been installed in the ML4100, then B Memory can also be uploaded and downloaded. To upload or download B Memory, cursor to the Action column and depress the f3 key ("UPLOAD B") or the f4 key ("DOWNLOAD B"). The B Memory is uploaded or downloaded to B Memory rather than a B Memory file, so the location of the cursor in the Action column is unimportant. The B Memory may then be saved in a B Memory file, if desired.

Figure 22a
B MEMORY STORAGE SCREEN
(Cursor in Action field)

SAVED B MEMORY DIRECTORY

B MEM	NAME	STATUS	WIDTH	ACTION
1	PRODM1	PROTECTED	32	RECALL
2	PRODM2	PROTECTED	32	RECALL
* 3 M	PRODM3	PROTECTED	32	RECALL
4	PRODM4	PROTECTED	32	RECALL
5		AVAILABLE		----
6		AVAILABLE		----
7	GLITCH	UNPROTECTED	4	SAVE
8	GLTCH2	UNPROTECTED	16	SAVE

CURSOR TO ACTION FIELD AND USE f KEYS
f1 = SAVE B f2 = RECALL B
f3 = UPLOAD B F4 = DOWNLOAD B
X=COPY A(record) to B(reference) - unsaved
B MEMORY CONFIGURATION (1,2.4 OR 8): 8
*=B memory in use M=memory modified

d. **B Memory File Names** (See Figure 22b)

B Memory files can be assigned names of up to six characters, before saving the file. To enter a B Memory file name, cursor to each space in the Name field and enter one character at a time, using the f1 and f2 keys to alphabetically advance or retreat characters until the desired character is displayed in each space. The f3 key (clear field) may be used to clear the entire name field.

To change a B Memory file name, edit the name field one character at a time, using the f1 and f2 keys, then save the file again. If a file name is changed without the file being resaved, a warning message will be issued if an attempt is made to exit the screen.

Figure 22b
B MEMORY STORAGE SCREEN
(Cursor in Name field)

SAVED B MEMORY DIRECTORY

B MEM	NAME	STATUS	WIDTH	ACTION
1	PRODM1	PROTECTED	32	RECALL
2	PRODM2	PROTECTED	32	RECALL
* 3 M	PRODM3	PROTECTED	32	RECALL
4	PRODM4	PROTECTED	32	RECALL
5	_____	AVAILABLE		----
6	_____	AVAILABLE		----
7	GLITCH	UNPROTECTED	4	SAVE
8	GLTCH2	UNPROTECTED	16	SAVE

CURSOR TO NAME FIELD AND USE f KEYS
f1=advance character f2=retreat character
f3=clear name
X=COPY A(record) to B(reference) - unsaved
B MEMORY CONFIGURATION (1,2,4 OR 8): 8
*=B memory in use M=memory modified

e. Status (Protected, Unprotected, Available) (See Figure 22c)

B Memory files normally have a status of "Unprotected," but may also be "Protected." To protect a B Memory file, cursor to the Status field, depress the f3 key (protected), then save the file. When the file is saved, a security code will be requested; this is a 2-digit hexadecimal number which is entered on the keypad. (If more than two keys are depressed, only the last two digits are used.) This security code will not be displayed on the screen.

This code will be requested prior to any future attempt to write over the file. If a security code is used to protect files, it is strongly recommended that it be written down somewhere.

To change a Protected file to Unprotected, cursor to the Status field, depress the f2 key (unprotected), and save the file again, using the correct security code. To erase a B Memory file, cursor to the Status field and depress the f1 key (available), and save the file again.

Figure 22c
B MEMORY STORAGE SCREEN
(Cursor in Status field)

SAVED B MEMORY DIRECTORY

B MEM	NAME	STATUS	WIDTH	ACTION
1	PRODM1	PROTECTED	32	RECALL
2	PRODM2	PROTECTED	32	RECALL
* 3 M	PRODM3	PROTECTED	32	RECALL
4	PRODM4	PROTECTED	32	RECALL
5	_____	AVAILABLE		----
6	_____	AVAILABLE		----
7	GLITCH	UNPROTECTED	4	SAVE
8	GLTCH2	UNPROTECTED	16	SAVE

CURSOR TO STATUS FIELD AND USE f KEYS
f1 = available f2 = unprotected
f3 = protected
X=COPY A(record) to B(reference) - unsaved
B MEMORY CONFIGURATION (1,2,4 OR 8): 8
*=B memory in use M=memory modified

3. COMPARING DATA (AB-032 only) (See Figs. 23a, 23b, 23c, and 23d)

The Compare screen is used to compare A (record) Memory to B (reference) Memory, using one of three methods:

* A Single Compare of already-captured data in A Memory to B Memory.
* A Continuous Compare where data is recaptured according to the machine setups, and compared to B Memory after each recapture.
* An Autosequencing Compare, which is similar to a Continuous Compare, except that up to 8 different machine setups and up to 8 different B memories (and thus up to 8 different Compares) are performed a selected number of times.

To execute any Compare, first use the Compare screen to specify several basic Compare parameters. Then, to execute a Compare, depress the f2, f3 or f4 key; results will be displayed on the screen.

To clear the results of a Compare, either depress the CLEAR key while in the Compare screen, or depress the START key to recapture data. To abort a Compare, depress the STOP key.

a. Single Compare (See Figure 23a)

Before comparing data, the user must select several comparison parameters:

* Data ranges for both A and B Memories (by specifying the A Memory Start Address, the B Memory Start Address, and the B Memory End Address)
* Comparison Mode (bit-to-bit or word-to-word)
* Skew Tolerance (jitter)
* Comparison Condition (A≠B or A=B)

Before performing a Single Compare, there must already be captured data in A Memory and reference data in B Memory. The data in B Memory may consist of previously captured data that was copied to B Memory with the Copy command, a B Memory file that was recalled from the Saved B Memory Directory, or data which was downloaded via the RS-232C interface.

To specify the data ranges for A and B Memories, cursor to the Address fields and enter the A Memory Start Address, the B Memory Start Address, and the B Memory End Address. The two ranges of data to be compared must be the same size. (The A Memory End address will be calculated and displayed automatically after the other three addresses are entered.)

There are two modes of compare: (1) Bit compare, in which words are compared on a bit-to-bit (channel-to-channel) basis, and (2) Word compare, in which words are compared on a word-to-word basis. To specify the comparison Mode, cursor to the Mode field and depress either 0 (bit) or 1 (word).

The maximum amount of skew allowed is +/- 9 words. Skew allows the Compare algorithm to be more flexible in comparing A to B Memories. For example, microprocessor cycles sampled with the internal clock may appear a varying number of times in the buffer, depending upon how the internal clock and the microprocessor clock happened to line up at a given time. Allowing some skew in the Compare lets the Compare algorithm look up to skew words on either side of the word in reference memory to try to find a match.

The Skew value will directly affect the amount of time it takes to compare A and B Memories. As the value of allowable skew goes up, the number of checks that the machine must perform increases dramatically. For faster execution times, it is recommended that the Skew value, along with the range of addresses to compare, be kept no larger than is absolutely necessary.

To specify Skew Tolerance, cursor to the Skew Tolerance field and enter a value between 0 and 9.

To specify the Compare Condition, cursor to the Condition field and depress either 0 (to select A≠B) or 1 (to select A=B). (When it is desired to compare A and B Memories for differences, specify A≠B; for matches between A and B Memories, specify A=B. Occurrences of the differences or matches will be then be counted and highlighted in the state display.)

To execute a Single Compare (after specifying the above parameters), depress the f2 key (Single Compare). This will cause the screen to display the Compare Results, as shown in Figure 23b.

Figure 23a
COMPARE Screen

```
COMPARE:              A MEMORY   B MEMORY
Start address:         +00000     +00000
End address:           +00124     +00124
Mode (0=bit, 1=word)            : word
Skew tolerance (in samples)     : +0
Condition (0=A≠B, 1=A=B)        : A=B

CONTINUOUS COMPARE:
Mode (0=halt if, 1=count if): count if A=B
Pass if (0=A≠B, 1=A=B)      : A=B
AUTOSEQUENCING:
Memory to use: 1  Passes to run: 00100

f1=toggle +/-          f3=continuous compare
f2=single compare      f4=begin autosequence
```

Figure 23b
COMPARE Screen
(Single Compare Results)

```
COMPARE:              A MEMORY   B MEMORY
Start address:         +00000     +00000
End address:           +00124     +00124
Mode (0=bit, 1=word)              : word
Skew tolerance (in samples)       : +0
Condition (0=A≠B, 1=A=B)          : A=B

          COMPARE RESULTS
NUMBER OF OCCURRENCES (A=B)=00125
FIRST OCCURRENCE=+00000
LAST OCCURRENCE =+00124


f1=recapture and compare
Press STATE to view results
Press CLEAR to clear results
```

After a Single Compare, the number of occurrences of the compare condition, along with the address of the first and last occurrences, are displayed on the Compare screen (Figure 23b above). The occurrences of the compare condition are highlighted in the STATE display.

To clear the results of a Compare, either depress the CLEAR key while in the Compare screen, or depress the START key to recapture data. To recapture data and perform another Compare, depress the f1 key. To abort a Compare, depress the STOP key.

It is often necessary to compare only selected fields in A and B Memories. This can be done by using B Memory edit to fill the fields to ignore with "Don't Cares", or by accessing the Format screen and turning off the fields to ignore. Turning off fields does not destroy data in B Memory. However, once B Memory has been edited to Don't Cares, there is no way to "un-Don't Care" the data; it must be saved and recalled from B Memory to restore the areas set to Don't Cares.

b. Continuous Compare (See also Section 3.a above)

Continuous Compare is helpful in finding an intermittent bug that only seems to show itself when no one is looking for it. Catching such a failure by performing Single Compares over and over again until it happens to be found is tedious; instead, the Continuous Compare feature can be used to direct the ML4100 to do the looking itself, then either stop when it catches the failure or else count the number of times it sees it.

To perform a Continuous Compare, first specify the addresses, Compare mode, skew tolerance, and Compare condition as described in the Section 3.a above (Single Compare). Then specify the Continuous Compare mode (whether the ML4100 should "Halt If" the condition is seen, or just "Count If" the condition is seen) and the pass condition (A≠B or A=B). For example, if there is a known good "picture" in B Memory, then A=B would be a "Pass"; conversely, if a known bad picture has been seen (possibly a recurring fault found in production test), then A=B is a failure, and the pass condition would be A≠B.

To specify the Continuous Compare mode, cursor to the Continuous Compare Mode field and depress either the 0 key ("Halt If") or the 1 key ("Count If"). To specify the pass condition, depress either the 0 key (A≠B) or the 1 key (A=B). To execute a Continuous Compare after specifying all of the necessary parameters, depress the f3 key. The results will be displayed, as shown in Figure 23c below. To abort a Continuous Compare, depress the STOP key.

Figure 23c
COMPARE Screen
(Continuous Compare Results)

```
COMPARE:              A MEMORY   B MEMORY
Start address:         +00000     +00000
End address:           +00124     +00124
Mode (0=bit, 1=word)            : word
Skew tolerance (in samples)     : +0
Condition (0=A≠B, 1=A=B)        : A=B

CONTINUOUS COMPARE:
Mode (0= halt if, 1=count if): count if A=B
Pass if (0=A≠B  1=A=B)       : A=B


PASS: 00178  FAIL: 00000   ATTEMPTS: 00178


COMPARING  - PRESS STOP TO ABORT
```

The results of a Continuous Compare (number of passes, fails, and attempts) will be continuously displayed and updated on the Compare screen. If the "Count If" Continuous Compare mode was specified, the compares will be executed forever, or until the STOP key is depressed. If the "Half If" mode was specified, the compares will be performed only until the condition specified has been met or until the STOP key is depressed.

If a Continuous Compare halts itself, to display the results of the last Compare that caused the halt, depress the CLEAR key once. As in a Single Compare, these results are highlighted in the State display. To clear the Compare results, depress the CLEAR key a second time.

Note that the Compare Condition to halt on and the condition specified as a failure (inverse of the Pass Condition) are two separate things; the Compare does not halt on a failure, but halts when the Compare Condition is met.

The Pass, Fail, and Attempt counters roll over to zero at 65535.

c. Autosequencing (See Figures 23a and 23d)

Autosequencing was designed with production test in mind. It is a very powerful feature which allows up to eight totally unrelated Compares to be made between captured data and saved reference data.

To perform autosequencing, first specify the parameters detailed in both Section 3.a (Single Compare) and Section 3.b (Continuous Compare), using the Compare screen. Then specify two more parameters (also using the Compare screen): the (B) Memory To Use for Compare (an already saved B Memory file) and the number of Passes To Run.

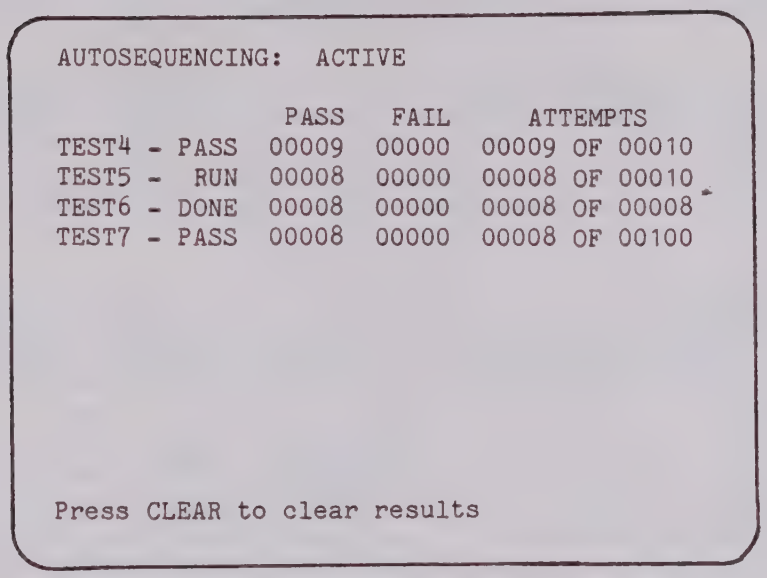
To specify the (B) Memory To Use, cursor to the Memory To Use field and enter the B Memory file number. To specify the number of Passes To Run, cursor to the Passes To Run field and enter the number (between 0 and 65535).

Next select the number of different Compares to perform by specifying the Autosequencing Start and End Files, using the Setup Directory screen. For example, to run four different tests requiring capturing and comparing data at four different locations, set up the ML4100 to trigger at the first location and capture data to use as a reference memory. Then save this data in a B Memory file for later use, and also save the machine setups used to capture this data in a setup file.

If the user has selected setup file number 4 and file number 1 for the B Memory data, then this process should be repeated 3 more times, saving the setup parameters used in setup files 5, 6 and 7, and the B Memory pictures in B Memory files 2, 3 and 4. Then the selected Autosequencing Start File would be setup file 4, and the End File would be setup file 7. (Note that setup files to use in Autosequencing must be in consecutive order; B Memories, however, may be in any order.)

To execute Autosequencing, depress the f4 key while in the Compare screen. Autosequencing results will be displayed and updated continuously on the Autosequencing Results screen (Fig. 23d); this screen is displayed automatically when an Autosequence is begun. To abort Autosequencing, depress the STOP key.

Figure 23d
AUTOSEQUENCING RESULTS


AUTOSEQUENCING: ACTIVE
PASS FAIL ATTEMPTS
TEST4 - PASS 00009 00000 00009 OF 00010
TEST5 - RUN 00008 00000 00008 OF 00010
TEST6 - DONE 00008 00000 00008 OF 00008
TEST7 - PASS 00008 00000 00008 OF 00100
Press CLEAR to clear results

As Autosequencing begins, setup file number 4 is read; it specifies the B Memory (file number 1) to load to use for the Compare. Data is captured, the Compare is made, and the progress of Test 4 is updated. Next, setup file number 5 is read. It specifies a B Memory file (file number 2) of its own to use as reference data. The machine captures new data, performs a new Compare, and continues. Each Autosequencing test runs until the Number of Passes to Run (specified in the Compare screen of each setup file) has been executed.

Test numbers correspond to the setup files in which the setups are stored. The number of passes, fails and attempts for each test is continuously updated, and the status of each test is displayed.

"RUN" means that the test is currently being run; "PASS" means that the last time the test was run, it passed; "FAIL" means that the last time the test was run, it failed; "DONE" means that the test has been run the number of passes requested and is now finished; and "HALT" means that Autosequencing has been halted because this test has encountered a "Halt If" condition.

To print a hard-copy record of the Autosequencing results (only if the RS-232 Communications Control Card or the RE-016 ROM Emulator Module is installed and the logic analyzer is a ML4100B), depress the f4 key. (There is no on-screen prompt for this print function of the f4 key.)

4. SEARCH (AB-032 only)

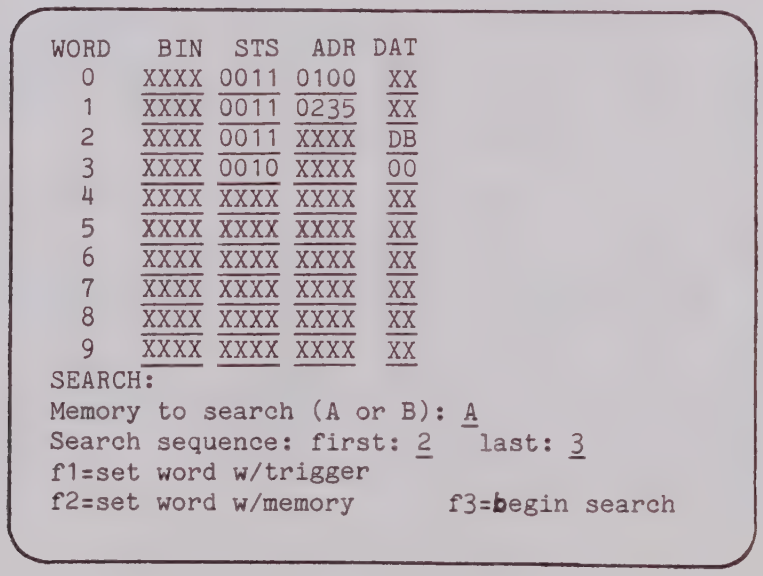
The Search screen is used to search for a pattern of 1 to 10 words labeled 0-9) in either A or B Memory. To define each word to be searched for, cursor to each of its fields and enter it via the numeric, X, and CLEAR keys on the keypad.

To specify which Memory to search, cursor to the Memory To Search field and enter A or B. To define the pattern to search for, cursor to the Search Sequence fields and enter the numbers of the first and last words which define the pattern to search for. (To search for a single word, enter the same number for both the first and last word.) To execute the Search, depress the f3 key.

A complex word from a trigger word or from Memory may be loaded into the search list. To fill a search word with a trigger word, position the cursor in the word to fill, and depress the f1 key (Set Word with Trigger); then select the trigger word to use (A, B, C, or D) and depress the f1 key to execute the fill.

To fill a search word with a Memory location, first cursor to the Memory To Search field and select either A or B (the Memory in which the word is located); next position the cursor in the word to be filled, and depress the f2 key. Then (using the cursor, numeric, f1 and CLEAR keys) edit the address of the memory location to use, and execute the fill by depressing f2.

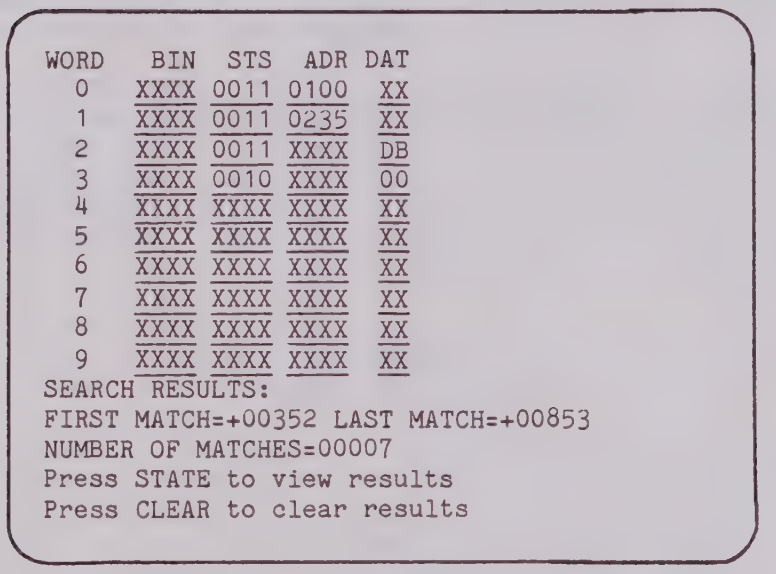
Figure 24a
SEARCH SCREEN


WORD BIN STS ADR DAT
0 XXXX 0011 0100 XX
1 XXXX 0011 0235 XX
2 XXXX 0011 XXXX DB
3 XXXX 0010 XXXX 00
4 XXXX XXXX XXXX XX
5 XXXX XXXX XXXX XX
6 XXXX XXXX XXXX XX
7 XXXX XXXX XXXX XX
8 XXXX XXXX XXXX XX
9 XXXX XXXX XXXX XX
SEARCH:
Memory to search (A or B): A
Search sequence: first: 2 last: 3
f1=set word w/trigger
f2=set word w/memory
f3=begin search

As soon as a Search has been executed, the Search results will be displayed on the Search screen. For example, Figure 24b shows the results of a Search in A Memory for an input from Port 0 (opcode fetch of a DB followed by a memory read of a 00). The number of search matches found, along with the addresses of the first and last matches, are shown on the screen. To view these matches highlighted on the State display screen, depress STATE.

To clear the results of a Search, either depress CLEAR while in the Search screen, or depress START to capture new data.

Figure 24b
SEARCH SCREEN
(Results)


WORD BIN STS ADR DAT
0 XXXX 0011 0100 XX
1 XXXX 0011 0235 XX
2 XXXX 0011 XXXX DB
3 XXXX 0010 XXXX 00
4 XXXX XXXX XXXX XX
5 XXXX XXXX XXXX XX
6 XXXX XXXX XXXX XX
7 XXXX XXXX XXXX XX
8 XXXX XXXX XXXX XX
9 XXXX XXXX XXXX XX
SEARCH RESULTS:
FIRST MATCH=+00352 LAST MATCH=+00853
NUMBER OF MATCHES=00007
Press STATE to view results
Press CLEAR to clear results

5. B MEMORY EDIT (AB-032 only) (See Figure 25)

With this screen, the user can edit reference data (in B Memory). A typical use might be to enter Xs (Don't Cares) in some positions in some fields so that a Compare will ignore extraneous data. As with all ML4100 data displays, the user can direct the cusor to any desired address in the buffer via the f1 key, and the cursor speed can be changed via the f2 key.

To edit B Memory data, use the f1 key and the Up and Down arrow keys to position the word to edit in the highlighted edit window. Use the numeric, X, left and right cursor, and CLEAR keys to edit the data to the desired value.

A range of addresses may be filled with a particular word or portion of a word. To fill a range of addresses with a word, first position one address of the range to fill (either Start or End Address), in the edit window, and depress the f3 key; then edit the data to the desired word to fill with. Next position the other address of the fill range in the edit window, using the Up and Down arrows or the f1 key, then depress the f3 key to execute the fill.

To fill a range of addresses with a portion of a word, first depress the FORMAT key and change the display format to selectively turn off the fields that are to be left undisturbed by the fill. Next execute the fill as described above, then return to the Format screen and turn all fields back on again. Note: One disadvantage of this method is that portions of trigger words are reset to Don't Cares; this problem can be alleviated by recalling a setup file to restore the original trigger words.

Figure 25
B MEMORY EDIT SCREEN

```
STATE  BIN  STS  ADR DAT
00007 0000 0011 0006  21
00008 0000 0010 0007  55
00009 0000 0010 0008  55
00010 0000 0011 0009  36
00011 0000 0010 000A  04
00012 0000 0001 5555  04
00013 0000 0011 000B  DB
00014 0000 0010 000C  00
00015 0000 0010 0000  FF
00016 0000 0011 000D  E6
00017 0000 0010 000E  01
00018 0000 0011 000F  CA
B EDIT: Cursor to position and edit data
f1=position cursor     f2=cursor spd: slow
```

6. B MEMORY DISPLAY

B Memory is displayed only on the State screen. A highighted field at the bottom of the State display screen indicates whether A or B Memory is being displayed. To display A or B Memory, access the State display screen and depress either A or B to select the Memory to be displayed.

As with A Memory, a pointer indicates the last displayed location in B Memory. When the State display screen is exited and re-entered, this pointer "remembers" the last area of Memory displayed. In addition, as the cursor is moved through one memory, the pointer to the other Memory not being displayed is moved at the same time. If a corresponding address does not exist in the other Memory, the pointer is left undisturbed.

To print a hard-copy record of any portion of A or B Memory (in an ML4100B with either the RS-232C Communications card or the ROM Emulator Module, RE-016) depress the f4 key while viewing the State display screen. There is no on-screen prompt for this print function of the f4 key.

C. EXECUTION TIMES

1. SETUP STORAGE TIMES

The time required to save a setup file is directly proportional to the difference, in number of bytes, between the current machine setups and the old file to write over. This time is at most 2.5 seconds, but on the average is less than 1 second.

2. B MEMORY STORAGE TIMES

The time required to save a B Memory file is directly proportional to the difference, in number of bytes, between the current B Memory and the old B Memory file to write over. In addition, the number of bytes in B Memory varies according to the B Memory configuration.

Storage times are:

Configuration	Max. Storage Time
1	20.0 seconds
2	10.0 seconds
4	5.0 seconds
8	2.5 seconds

3. COPY A TO B EXECUTION TIMES

The time required to copy A Memory to B Memory is directly proportional to the number of samples to copy. Number of samples is dictated by the width of the machine.

Width	Approx. Copy Time
4	15.0 seconds
8	7.5 seconds
16	4.3 seconds
32	2.6 seconds

4. SEARCH EXECUTION TIMES

The time required to search a memory depends on the number of words to search (width of machine), the size of the memory (B Memory only) and, to a lesser degree, the number of matches found and the size of the search pattern.

a. A Memory

Width	Approx. Search Time
4	16.0 seconds
8	8.0 seconds
16	4.0 seconds
32	2.0 seconds

b. B Memory

	Approx. Search Time (in sec.)			
	Configuration			
Width	1	2	4	8
4	5.0	2.5	1.3	0.6
8	2.5	1.3	0.6	0.3
16	1.3	0.6	0.3	0.2
32	0.8	0.4	0.2	0.1

5. COMPARE EXECUTION TIMES

The time required to compare A Memory to B Memory is dependent upon the size of the Compare range, the allowable skew in the Compare, and, to a lesser degree, the Compare mode (word or bit) and the number of Compare occurrences found. Note also that, in Continuous or Autosequencing Compares, the machine may not need to compare the entire memories if "Count If" is selected. Sample Compare times are:

Approx. 3.5 seconds for:	32 bits wide Word compare 0 skew 1000 samples No occurrences found
Approx. 135 seconds for: (Worst case)	4 bits wide Bit compare +/-9 skew 8000 samples No occurrences found

To keep Compare execution times to a minimum, keep the size of the Compare range and the allowable skew only as large as necessary. This is especially important in Continuous and Autosequencing Compares, since the Compare time represents "dead" time regarding catching a failure. A typical Compare of microprocessor data needs a Compare range of 100 samples and 0 skew; this Compare would take less than 1 second.

(blank)

X. 16M-680 MICROPROCESSOR POD

(Supports Motorola 68000/68010 microprocessors)

A. GENERAL

1. SPECIAL REQUIREMENTS FOR ML4100 TO ACCOMMODATE 16-BIT MICROPROCESSOR PODS

B Model: Early models of the ML4100 (those shipped before March 16, 1984) must be upgraded to an ML4100B by the factory in order to accommodate a 16-bit microprocessor pod.

Revision L Base ROMs: The ML4100B base firmware must be Revision L or later. (If so, the top of the Status screen will be labeled "LB" or later.)

DI-012 Disassembly ROM Board: A DI-012 Disassembly ROM Board must be installed on the ML4100B. This board accommodates several microprocessor disassemblers, and supports both 8-bit and 16-bit microprocessor pods.

62-Bit Trigger ROM: A 62-bit trigger ROM (which is labelled "MT600" and is supplied with all 62-bit wide microprocessor pods) must be installed on a DI-012 ROM Disassembly Board inside the ML4100B.

2. OVERVIEW OF 16-BIT MICROPROCESSOR POD OPERATION

The 16M-680 Microprocessor Pod clips directly over a 68000 or 68010 DIP. The pod collects data (at up to 62 channels) at microprocessor clock speeds of up to 16 MHz, and formats it for logic analysis and full program disassembly (including all exceptions and bus cycles) by the ML4100B.

The pod has three operating modes (Filtered, Transparent, and Jump), and produces disassembled code in each of them. Special hardware in the pod tracks every bus cycle of the microprocessor and determines if the particular data will be recorded in the pod's current operating mode. In the Filtered mode, all unused prefetch cycles are removed in the pod hardware so they won't activate unwanted trigger events and won't be stored in the ML4100B memory. In the Transparent mode, all bus cycles are passed along to the ML4100. In the Jump mode, all bus cycles are filtered out except for changes in program flow (branches, subroutine calls, exceptions, etc.), providing a disassembled "map" of how the code was actually executed.

The Disassembly display details all microprocessor data cycles ("unscrambled" and put back together with their corresponding instructions), provides labels for readability, and marks all instructions as either "u" for User or "s" for Supervisor, depending upon how they are fetched. The State display shows data as it occurs on the address and data busses. Data may also be viewed on the Timing display; however, there is no internal clocking capability, and only in the Transparent or Jump modes are samples stored in strictly increasing time order. (Filtered mode delays the storage of instruction cycles relative to that of data cycles.)

If the microprocessor ceases operation (e.g., because of a HALT condition or a lack of DTACK, BERR, or VPA acknowledge during a bus cycle), several preceding bus cycles will remain stored in the pod. These cycles cannot be recorded or triggered upon until the microprocessor resumes operation.

B. HARDWARE

1. SWITCH SETTINGS

On one end of the 16M-680 pod there is an access hole for a four-position DIP switch. This access hole is on the left end of the pod when viewed with the label lettering upright.

Switch 1 selects the processor type:

Down = 68000

Up = 68010 (To select the 68010, Switch 3 must also be Up.)

NOTE: ML4100B power must be cycled (turned off, then on again) whenever the position of Switch 1 has been changed.

Switch 2 selects the data source for Probes 30, 29, & 28 (the leftmost three significant bits of the BIN field on the State display):

Down = the Interrupt Priority Level lines on the processor (IPL2*, IPL1*, & IPL0*, active low, are Probes 30, 29, & 28, respectively.)

Up = the external clip leads (Install the 20-pin lead set from the LP-320 32-Channel Logic Pod in the 20-pin connector on the right end of the pod; Probes 30, 29, and 28 then correspond to Leads 02, 01, and 00.)

NOTE: The position of this switch may be changed during use; i.e., the power needn't be cycled (turned off, then on again).

Switch 3 selects the filtering type:

Down = 68000 systems without wait states or DMA cycles

Up = 68010, & 68000 systems with wait states or DMA cycles

The effect of this switch is complicated and is explained in Section D, "Filtered Mode."

NOTE: The position of this switch may be changed during use; i.e., the power needn't be cycled (turned off, then on again).

(Switch 4 is unused in this pod.)

2. SIGNAL ASSIGNMENTS

Fifty-four signals are saved in each cycle of trace memory:

16 Data lines
24 Address lines (The least significant address line is created by hardware in the pod.)
14 Discrete lines (One of them is used for test and has no meaning to the user)

The data is saved in trace memory inverted, so the sense of the data displayed is automatically set to INV on the Format screen. The probes are displayed in four groups (Address, Data, Status, and other Binary). The following tables relate probe numbers to signals and describe the signals.

Figure 26
SIGNAL ASSIGNMENTS--OVERVIEW

Type	ML4100 Probe Assign-ments	Default Positions on Format Screen (See Fig. 27)	Significance
	63	-	(Not available for use)
Status (STS)	62-56	5	Processor Status (Cycle ID)
Address (ADR)	55-32	6	Processor Add. lines 0-23
	31	-	(Not available for use)
Binary (BIN)	30-24	1	(See Binary Field, below)
	23-16	-	(Not available for use)
Data (DAT)	15-0	7	Processor Data lines 0-15

Figure 27
DISPLAY FORMAT SCREEN

```
                    DISPLAY  FORMAT

TIMING:
Line     1  2  3  4  5  6  7  8  9 10 11 12
Probe   00 01 02 03 04 05 06 07 XX XX XX XX

STATE:                              STS ADR DAT
Field          1    2    3    4    5    6    7    8
Base         BIN  OFF  OFF  OFF  BIN  HEX  HEX  OFF
High Probe    30   00   00   00   62   55   15   00
 Low Probe    24   00   00   00   56   32   00   00
Data Sense   INV  INV  INV  INV  INV  INV  INV  INV
Bases: A=ASC B=BIN C=OCT D=DEC E=HEX F=OFF
Senses: 0=NON (non-inverted), 1=INV

POD MODE (also sets defaults): 68000T
f1=68000T  f2=68000F  f3=68000J
```

Figure 28
SIGNAL ASSIGNMENTS--DETAILED

(Asterisk indicates active low)

PROBE NO.:	SIGNAL:
Binary (BIN) -- (field of 7 binary numbers)	
30	IPL2*, or Lead 02 if external signals are used
29	IPL1*, or Lead 01 if external signals are used
28	IPL0*, or Lead 00 if external signals are used
27	HALT*
26	(Used for test)
25	VMA*
24	BERR*
Status (STS) -- (field of 7 binary numbers)	
62	FETCH* or SOURCE, depending on Filtered or Jump mode (generated by the pod)
61	BGACK*
60	BYTE
59	R/W*
58	FC2
57	FC1
56	FC0
Address (ADR) -- (field of 6 hexadecimal numbers)	
55-52	A23-A20
51-48	A19-A16
47-44	A15-A12
43-40	A11-A08
39-36	A07-A04
35-32	A03-A00
Data (DAT) -- (field of 4 hexadecimal numbers)	
15-12	D15-D12
11-8	D11-D08
7-4	D07-D04
3-0	D03-D00

To change the format of the State display (including number base), see Section III.H.2, "Format for State Display Mode." Default number bases for the State display are:

Figure 29
DEFAULT NUMBER BASES FOR DISPLAYS

Field	Base
Binary (BIN)	Binary
Status (STS)	Binary
Address (ADR)	Hexadecimal
Data (DAT)	Hexadecimal

Figure 30
QUICK REFERENCE LIST FOR
SOME STATUS & BINARY SIGNALS

BGACK*	When 0, indicates nonprocessor cycles, such as DMA
BTYE	When 1, indicates 8-bit transfers
R/W*	When 0, indicates Write; when 1, indicates Read
VMA*	When 0, indicates autovectored interrupts or special hardware interface, e.g., 68000 peripheral
BERR*	When 0, indicates bus error
HALT*	When 0, and BERR* is also 0, indicates cycle is to be retried
FETCH*	When 0, indicates the cycle is the first word of an instruction (Filtered mode only)
SOURCE	When 1, indicates the cycle is the source of a program discontinuity (the "from" address) (Jump mode only)

Figure 31
DETAILED LIST OF
BINARY AND STATUS SIGNALS

(Asterisk indicates active low)

Probe 62: FETCH* is created by the pod in the Filtered mode; it is asserted during the first cycle of an instruction.

Probe 62: SOURCE is created by the pod in the Jump mode. It is high if the cycle is a Source, and low if it is a Destination; if the cycle is both a Destination and a Source, the bit will be low. If in Filtered mode, this bit indicates the first word of an instruction fetch.

Probe 61: BGACK* is the Bus Grant Acknowledge input to the processor. When low, cycles are performed by a device other than the processor. In Filtered mode, such cycles are removed from the data stream; in Transparent mode, any BGACK* cycles in the buffer are ignored by the disassembler.

Probe 60: BYTE is created by pod hardware; it is high when only one data strobe is asserted.

Probe 59: R/W* is the READ/WRITE* signal from the processor; it is high during Read cycles and low during Write cycles.

Probes 58-56: FC2, FC1, and FC0 are the processor function codes which indicate which of several address spaces is accessed by a given cycle. (These "spaces" are listed Section 3, below, "Status Codes.")

Probe 32: A0 is created by pod hardware; it is high when the lower data strobe (LDS) is asserted and the upper data strobe (UDS) is negated.

Probes 30-28: Lead 02, Lead 01, & Lead 00, or IPL2*, IPL1*, & IPL0* (Interrupt Priority Level lines), depending upon whether the position of Switch 2 on the pod is Up or Down, respectively. (See Sec. B.1, above, "Switch Settings.")

Probe 27: HALT* -- See Probe 24, BERR*.

Probe 26: Used for test, and undefined for normal operation.

Probe 25: VMA* is the Valid Memory Address output from the processor, asserted in response to the VPA* (Valid Peripheral Address) signal. These signals are used primarily when interfacing with 6800 peripherals. If VPA* is asserted by external hardware during an Interrupt Acknowledge cycle, the processor performs an autovector operation. Note that data sampling is different during VMA* cycles, using the falling edge of E rather than the rising edge of the data strobes. (See microprocessor data sheet for details.)

Probes 24, 27: BERR* and HALT* signals are taken directly off the processor. When the BERR* signal is asserted without HALT*, the processor takes a bus error exception; when BERR* and HALT* are both asserted, the processor will retry the cycle. In the Filtered mode, cycles with both BERR* and HALT* asserted (cycles to be retried) are removed, as the cycle will recur.

3. STATUS CODES

The BIN (Binary) and STS (Status) fields (on the default Trigger screen and State display screen) contain status information about the bus cycle. The three least significant bits of the STS fields are the function codes FC2, FC1 and FC0 of the processor, which define the "space" of the cycle as shown:

FC2	FC1	FC0	
0	0	1	User Data space
0	1	0	User Program space
1	0	1	Supervisor Data space
1	1	0	Supervisor Program space
1	1	1	CPU space; used for Interrupt Acknowledge and Breakpoint cycles

(Other combinations are unassigned and cannot be accessed in the 68000; the 68010 can access them via the MOVES instruction.)

An abbreviated list of the BIN and STS signals is available on the TRIGGER screen, shown below.

Figure 32
TRIGGER SCREEN (WITH 16M-680 POD)

```
                    TRIGGER
          BIN      STS     ADR   DAT
A:   1111111  0101110  000300  3E3C

B:   1111111  1101110  000314  660A

C:   XXXXXXX  XXXXXXX  XXXXXX  XXXX

D:   XXXXXXX  XXXXXXX  XXXXXX  XXXX

Sequence:    A then B
STS: (6-0 represent bit position in field)
  6      5     4   3   2-0(Func Codes 2-0)
SOURCE BGACK* BYT R/W* 001 010 101 110 111
                       UsD UsP SuD SuP IAK
BIN: 6-4(see pod) 3=HALT*  1=VMA*  0=BERR*
```

Figure 33
STATUS CODE EXAMPLES

	STS
READ User Data Space Word	X101001
WRITE User Data Space Word	X100001
READ User Data Space Byte	X111001
READ User Program Space Word	X101010
READ User Program Space First Instruction Word (Filtered mode only)	0101010
READ Supervisor Program Space First Instruction Word	0101110
WRITE Supervisor Data Space Byte	X110101
Interrupt Acknowledge (The level of the interrupt being acknowledged is available on Address Lines A3, A2, and A1. Also, in the 68010, Interrupt Acknowledges must be distinguished from breakpoints by using upper address lines.)	X101111

	BIN	STS
Autovectored Interrupt Acknowledge (VMA* asserted in response to VPA*)	XXXXX0X	X101111
Bus Error Cycle	XXX1XX0	X1XXXXX
Retry Cycle (Transparent mode only)	XXX0XX0	X1XXXXX
VMA* Cycle (special hardware)	XXXXX0X	
Nonprocessor Cycle (DMA)	XXXXXXX	X0XXXXX

C. TRANSPARENT MODE

In the Transparent mode, all bus cycles (including unused prefetches) are sent to the ML4100B logic analyzer in the order in which they occur. Data captured in the Transparent mode can be viewed in all three display formats: State, Timing and Disassembly. The disassembler discards unused prefetches, so that disassembled data is the same as it is in the Filtered mode.

(See also Sections F.3, "Undesired Triggering on Prefetches in Transparent Mode," and D.2.b, "FIFO Memory.")

D. FILTERED MODE

1. GENERAL

As with most 16-bit microprocessors, the 68000 uses an instruction prefetch mechanism to enhance performance. While one instruction is being executed, the two words following it have already been fetched from program memory. If an instruction or exception causes a branch, one or both of the prefetched words will be discarded. With logic analyzers that fail to filter prefetched words, triggering on a location in program memory guarantees only that the instruction has been fetched, not executed.

To eliminate undesired triggering, the Arium 16M-680 Microprocessor Pod tracks every bus cycle and filters out unused prefetches from the instruction stream sent to the ML4100B Logic Analyzer. Increased data storage is an added benefit of filtering, because unused prefetches do not take up any space in the 500-cycle trace buffer of the ML4100B.

Data captured in the Filtered mode may be displayed in State, Timing and Disassembly formats.

2. HARDWARE CONSIDERATIONS

a. General

When in Filtered mode, special hardware in the 16M-680 Microprocessor Pod monitors the processor's bus cycles to determine when prefetched cycles have not been executed. Because as many as thirty bus cycles may occur between the read of an instruction and its execution (or its being discarded), the pod must temporarily hold the instruction off to one side of the data stream until it has enough information to determine whether the cycle has actually been executed.

The result is that (in Filtered mode) all instruction words appear in the ML4100B memory and in the State display where the next instruction word would have appeared if in Transparent mode, except for instruction words which are found to be unused prefetches, which are removed completely.

Note that, due to the delay of Filtered mode, the cycles appear in the ML4100B memory in the same order in which the processor actually performed them, not in the order in which they were fetched off of the bus by the prefetch mechanism. This may be important to the user when viewing external signals or interrupt levels on the State display.

b. FIFO Memory

In Filtered mode, a small (4-cycle), first-in-first-out memory in the pod removes cycles in the event of an interrupt, when two prefetched cycles are discarded. If the processor stops execution, this FIFO memory will contain the last four cycles and they will not have been captured in the ML4100B trace memory. (In Transparent mode, the length of this FIFO memory is reduced to one cycle.)

c. Unremoved Prefetches

Three instructions (CHK, DIV, and TRAPV) and the TRACE exception generate exceptions in a way that makes it difficult to remove all prefetches; in fact, it is sometimes impossible to know whether one or two prefetches need to be removed. In these cases only one prefetched word (the last) is removed.

Address and Bus Error exceptions do not have unused prefetches, but a cycle may be deleted anyway. The address of this cycle is pushed on the stack and labeled "AA" (see Section G.3.c.(4), "Exceptions"), so the loss of this cycle should not pose a problem.

Another ambiguous circumstance occurs when the processor executes a single--word conditional branch around a one-word instruction that makes no data accesses. The difficulty arises because the data stream is identical, regardless of whether or not the branch is taken. In the 68010, the execution speed is also identical, regardless of path, and so it is not possible to know whether the one-word instruction has been executed. The 68000 takes two clock cycles longer when the instruction has been executed than when it has not been executed, and the pod monitors this in order to decide whether or not the instruction has been skipped and is to be discarded. The clock cycle counting information may not be valid in systems with wait states or in which DMA cycles can occur. In either case, the one-word instruction following the branch is marked with a "?" in the disassembly display to indicate the uncertainty about its actual execution.

The usage of the cycle counting information is controlled by the user, via Switch 3 on the pod. For 68010 systems, this switch must be Up, or all such instructions will be removed. For most 68000 systems, this switch should be Down. For 68000 systems with DMA or wait states, the switch should usually be Up; the switch may be Down, but the user must be aware of the possibility of cycles being filtered out which have actually been executed.

d. Self-Synchronization

In order to make intelligent decisions about unused prefetches, the pod must synchronize itself with the processor. Under ordinary circumstances, only a few cycles are required for synchronization, and it will actually self-synchronize before the ML4100B begins recording data.

There are certain patterns of instructions for which the pod is not guaranteed to synchronize itself, but such patterns are virtually always tight loops of fewer than ten unusual instructions with no Writes. If such a pattern occurs, the pod may incorrectly remove cycles from the data stream, and the user must either use the Transparent mode, or restart data acquisition in the Filtered mode in hopes that the pod will synchronize itself. (Such self-synchronization may indeed occur, because synchronization may depend upon the exact point in the pattern at which acquisition begins.)

A processor RESET will cause the pod to lose sync. It should resynchronize within a few cycles, but for analyzing the processor's behavior at RESET, Transparent mode is recommended.

E. JUMP MODE

The Jump mode filters out all bus cycles, except those preceding and following events that cause changes in program flow (such as branches, subroutine calls, and exceptions). This feature is very useful for checking program flow and determining which sections of code are being executed. Data captured in the Jump mode can be viewed in all three display formats: State, Timing and Disassembly.

The Disassembly dispaly screen describes each entry as a Source ("was branched from") or Destination ("was branched to"). Since only the first word of each instruction is stored in the buffer, only the mnemonic of each instruction is displayed. If an instruction is both a Destination and a Source (e.g., when a jump lands at another jump), the instruction appears only as a Destination in the state buffer (i.e., no source entry is recorded, and there will be two Destination entries in a row). Thus, it is always possible to set a trigger on a Destination, but those Source instructions which are also Destinations will never be recorded with the "SOURCE" bit set in the STS field.

F. TRIGGERING CONSIDERATIONS

1. TRIGGERING SEQUENCE LIMITATIONS IN 62-BIT OPERATION

The basic width of the ML4100B data path is 32 bits. In order to accommodate 16-bit processors, the ML4100B divides a processor cycle into halves, and these halves are stored sequentially. (One bit of each set of 32 bits marks which half is which, leaving 31 bits per half and a total of 62 bits.)

The basic triggering mechanism in the ML4100B is also 32 bits wide and has up to 13 states. When only the upper 31 bits of information (Status and Address fields) are used, all triggering capabilities are available (as described in Sections III.E, III.F, and III.G). However, it is often necessary to use the Binary and Data fields to trigger. Special software allows the user to specify a single 62-bit trigger word.

In general, the software allows the user to be unconcerned with the triggering width of the machine; there are, however, some limitations. Some complex sequences required too many states to be implemented in the 13-state sequence machine. For example, the Trigger Sequence "A + B + C + D" requires 16 states to represent the condition past the first half of the trigger word, so this sequence is unavailable when Binary and Data fields (other than "Don't Cares") are specified. "Without" conditions also add complexity.

All of the ML4100's predefined sequences except "A + B + C + D" are available. For that sequence and for a User-Defined Sequence, the number of states needed for a given condition level can be calculated using these rules:

A single word requires two states. If further words are "ORed" in, double the number of states for each additional "true" word (e.g., A as opposed to A*). If words are "ANDed" in, double for each additional "not" word (e.g., A*).

Examples of number of states required for a given condition level are:

Condition	No. of States
A	2
A + B	4
A + B + C*	4
A & B	2
A* & B*	4
A + B + C	8
A* & B* & C*	8

Each level consists of two conditions; add the number of states required for each set of two conditions and multiply the resultant sum by the count for that level. Then sum at each level to obtain the total number of states.

However, it is usually not worth the effort to determine the number of states required by a particular sequence, because the 62-bit trigger and sequence software determines whether the specified sequence is too complex. Simply enter the sequence via the Trigger Sequence screen, and the ML4100B software will decide whether the sequence is too complex. If it is, the ML4100B will then display the Status screen with a flashing error message ("Trigger Too Complex") at the top; just remove a word from a condition or reduce the count at a level, and try it again.

2. DATA QUALIFICATION

Due to the 32-bit nature of the ML4100B capture and trigger circuitry, only the first 31 bits (the default Status and Address fields) of a processor cycle may be used to qualify data storage. Thus cycles may be qualified based on Address and access type, but not on the Data field. Although the qualification decision is based on the first (upper) half of the cycle, the same decision holds for the second (lower) half of the cycle; i.e., whole 62-bit frames are either qualified or not qualified.

3. UNDESIRED TRIGGERING ON PREFETCHES IN TRANSPARENT MODE

In the Transparent mode, prefetched instructions which are not executed may cause undesired triggering, i.e., triggering of the logic analyzer without the occurrence of the event of interest. Such unused prefetches are found before interrupts, after branches and returns, and sometimes after jumps and calls. Because these instructions have not been executed, they will not appear on the Disassembly screen.

To avoid undesired triggering, use the Filtered mode, which strips unused prefetches from the data stream (described in Section D, "Filtered Mode").

G. DATA DISPLAY

1. GENERAL

With the 16M-680 pod in Transparent mode, the State display shows all instruction fetches and data cycles in the same order in which they occurred on the bus. In Filtered mode, the order of cycles in the trace buffer will not match the actual order of cycles on the processor bus (see Section D, "Filtered Mode"), and any unused fetches (fetches following an instruction causing a program discontinuity) are disqualified by the pod and not stored in the trace buffer.

The Disassembly display groups these program fetches and data cycles into 68000 assembly language instructions. Regardless of whether or not the pod is filtering unused fetches, these unused fetches are not displayed on the Disassembly screen. The current pod mode is shown in a reversed field at the top of the Disassembly screen.

The formatting and display of data is described generally for the ML4100 in Section IV, "Operation." Further details of data display when using 16-bit microprocessor pods, including typical State and Disassembly display screens, keyboard functions, and display formats, follow.

Figure 34
STATE DISPLAY (ALL MODES)

```
STATE     BIN     STS    ADR  DAT
-0002 1111111 0111101 000350 2EFF
-0001 1111111 0101110 000316 B828
 TRIG 1111111 0101110 000320 51C8
00001 1111111 0101110 000322 FFF0
00002 1111111 0101110 000312 B618
00003 1111111 0101110 000324 1C80
00004 1111111 0101110 000326 0839
00005 1111111 0110101 000000 FFFF
00006 1111111 0101110 000328 0007
00007 1111111 0101110 00032A 0000
00008 1111111 0101110 00032C 0FFE
00009 1111111 0101110 00032E 6600
00010 1111111 0111101 000FFE 00FE
00011 1111111 0101110 000330 00D0
f1=position cursor  A  f2=cursor spd: slow
```

Figure 35
DISASSEMBLY DISPLAY
(TRANSPARENT AND FILTERED MODES)

```
STATE PC     DISASSEMBLY    68000T     S/U
 TRIG 000320 DBF     D0,000312          s
00003 000324 MOVE.B  D0,(A6)            s
      000000<FF
00004 000326 BTST    #07,00000FFE       s
      000FFE>00
00009 00032E BNE.L   000400             s
00012 000332 DBF     D7,000304          s
00014 000304 MOVEA.L #00000346,A0       s
00017 00030A MOVEM.W (A0),LIST          s
      LIST=D0-D6/A1-A5
      000346>2E61          000348>2E73
      00034A>2E63          00034C>2E69
      00034E>2E69          000350>2E2E
      000352>2E64          000354>2E65
f1=Pos. Cursor f2=Spd: slow f3=Cycles: on
```

Figure 36
DISASSEMBLY DISPLAY (JUMP MODE)

```
STATE PC     DISASSEMBLY    68000J     S/U
-0010 F00DB0 BEQ.S     (SRC)            s
-0009 F00DBE BTST      (DEST)           s
-0008 F00DB2 BEQ.S     (SRC)            s
-0007 F00DBE BTST      (DEST)           s
-0006 F00DB2 BEQ.S     (SRC)            s
-0005 F00DBE BTST      (DEST)           s
-0004 F00DB2 BEQ.S     (SRC)            s
-0003 F00DBE BTST      (DEST)           s
-0002 F00DB2 BEQ.S     (SRC)            s
-0001 F00DBE BTST      (DEST)           s
 TRIG F00DC8 RTS       (SRC)            s
00001 F03E90 TST.B     (DEST)           s
00002 F03E96 BSR.L     (SRC)            s
00003 F03DFE BSR.L     (DEST)           s
f1=Pos. Cursor f2=Spd: slow f3=Cycles: off
```

2. KEY FUNCTIONS ON THE DISASSEMBLY SCREEN

a. f1 Key (Position Cursor)

To move directly to any location in the buffer, depress the f1 key, enter the state number you wish to display, and depress f1 again. Negative state numbers correspond to cycles before the trigger event; positive state numbers correspond to cycles after the trigger. (If f1 is inadvertently depressed, depress it again to leave the cursor at its current location.)

b. Up/Down Arrow Keys (Scroll through Buffer)

To move through the buffer forward or backward, use the Up and Down Arrow keys.

c. f2 Key (Scroll Speed)

To scroll slowly through the buffer (one instruction at a time), use the f2 key to set the scroll speed to Slow. To scroll faster, toggle the f2 key to set the scroll speed to Fast.

d. f3 Key (Data Cycles On/Off)

To view the disassembled instructions without their data cycles, depress the f3 key. This gives a display very similar to an actual code listing and makes it easier to locate a particular section of code. To view the data cycle associated with the instruction, toggle the f3 key to turn the data cycles back on.

e. f4 Key (Print Screen)

To print a hardcopy report of the Disassembly screen, depress the f4 key. (There is no on-screen prompt for this function.) Note that the RS-232C option must be installed; this is included in the ROM Emulator Module, RE-016.

f. DISASM Key (Display Trigger Condition)

To display the trigger event, depress the DISASM key once to enter the Disassembly screen, and again to display the trigger event.

3. DISPLAY FORMATS

a. Instruction Display

As with 8-bit microprocessor pod disassembly, each instruction displayed contains a state number corresponding to the instruction position relative to the trigger event, the value of the Program Counter when the instruction was fetched (PC), and the mnemonic and operands for the instruction itself. Also, with the 68000/68010 pod, an "s" or "u" at the far right of the screen indicates whether the instruction was fetched from Supervisor space or from User space.

b. Data Cycles Display

Data cycles are displayed on lines following the instruction that caused them. These cycles may be turned on or off by toggling the f3 key.

Each data cycle contains an address, an indication of whether data was read from (>) or written to (<) the address, and the data itself. Each address consists of six hexadecimal characters (24 bits), while data may be a byte (2 characters), a word (4 characters), or a long word (8 characters).

Even though the 68000 performs a long word access by doing two word accesses, the ML4100B concatenates the data and shows it as as one data cycle on the Disassembly screen.

c. Specific Instruction Display Techniques

(1) MOVEM

The register list coded with the MOVEM instruction is reconstructed and shown as the second line of the instruction. This line is part of the instruction and stays displayed even with data cycles off. The data cycles associated with the MOVEM instruction are in the same order (reading left to right) as they are on the register list. (See Figure 35, "Disassembly Display, Transparent and Filtered Modes.")

(2) BSR, JSR

The data cycle showing the value of the Program Counter pushed onto the stack is labelled "PC:".

(3) RTD, RTS, RTR

The data cycle showing the value of the Program Counter popped from the stack is labelled "PC:".

(4) Exceptions

(a) PC, SR, VO, IR, AA Labels

The data cycles showing the values of the Program Counter and Status Register pushed onto the stack are labelled "PC:" and "SR:", respectively. If the processor is a 68010, the data cycle showing the Vector Offset is labelled "VO:". On a 68000 bus error or address error exception, the data cycle showing the Instruction Register push is labelled "IR:" and the cycle showing the Access Address push is labelled "AA:".

(b) 68010 Bus or Address Error Exception

On a 68010 bus error or address error exception, the following data cycles are labelled:

FA:	Fault Address
DO:	Data Output Buffer
DI:	Data Input Buffer
II:	Instruction Input Buffer

(c) Unlabelled Cycles

Unlabelled cycles correspond to Motorola internal information. (See Motorola M68000 Programmer's Reference Manual, M68000UM(A)D4, for additional information.)

(d) Interrupts

If the exception is an interrupt, then the type of interrupt (externally vectored, autovector, spurious or uninitialized), interrupt priority level, and interrupt vector number are displayed and labelled as such.

(e) Exception Vector

All exceptions label the exception vector which is read as "VECTOR:".

H. DISASSEMBLY SOFTWARE

The Disassembly software displays only executed (as opposed to prefetched) instructions, regardless of whether the pod is in Filtered mode (sending up only executed cycles) or Transparent mode (sending up all cycles).

To do this, the disassembly software must "sync up" to the cycles in the trace buffer, in much the same way as the pod hardware syncs up to the processor in Filtered mode. The disassembly software finds the first program space read, assumes it is an opcode fetch, and checks the following cycles to see if it made a correct assumption. If the following cycles don't match the expected pattern of instruction fetches, data cycles and prefetches for that particular opcode value, then the software assumes it found a subsequent byte of an instruction or a program space operand read. It ignores this cycle, finds the next program space read, and repeats the process.

This software process of syncing up to the trace cycles is performed in both Transparent and Filtered modes. This seems unnecessary in Filtered mode, because the pod sends up information identifying the first word of an instruction. Data qualification, however, may filter out subsequent cycles of an instruction, so that relying upon the fetch bit may be insufficient.

If the software cannot make sense of the cycles in the trace buffer, it will display the message "Can't disassemble--PS reads used as opcodes". It then shows disassembly of the buffer, assuming that every PS read was the first word of an instruction. Each line of display is marked with a "?" to indicate the software's uncertainty with the disassembed code.

If the software cannot make sense out of some of the cycles in the buffer, but then can resync to the trace cycles, it displays the message "Unable to disassemble XXX cycles," where "XXX" is the number of processor cycles that did not fit the expected pattern.

In addition, a special case exists when a one-word instruction with no data cycles follows a Bcc instruction. The processor cycle pattern appears identical, whether or not the branch has actually been taken. The software cannot determine whether or not the instruction has been executed, so it marks it with a "?".

(Blank)

XI. 16I-086 MICROPROCESSOR POD

(Fully supports the Intel 8086 and 8088 microprocessors, and the Intel 8087 coprocessor, including disassembly; partially supports the Intel 8089 coprocessor by collecting data, but without disassembly. If the pod is to be clipped onto an 8087 or 8089 directly, a hardware jumper inside the pod must be moved; see Section B.4, "Pod Connection Error Warning.")

A. GENERAL

1. SPECIAL REQUIREMENTS FOR THE ML4100 TO ACCOMMODATE THE 16I-086 POD

B Model: Early models of the ML4100 (those shipped before March 16, 1984) must be upgraded to an ML4100B by the factory in order to accommodate a 62-bit pod. (All pods for 16-bit microprocessors use 62-bit recording width.)

Revision L Base ROMs: The ML4100B base firmware must be Revision L or later. (If so, the top of the Status screen will be labeled "LB" or later.)

DI-012 Disassembly ROM Board: A DI-012 Disassembly ROM Board must be installed on the ML4100B. This board accommodates several microprocessor disassemblers, and supports both 8-bit and 16-bit microprocessor pods.

62-Bit Trigger ROM: A 62-bit trigger ROM (which is labelled "MT600" and is supplied with all 62-bit wide microprocessor pods) must be installed on a DI-012 ROM Disassembly Board inside the ML4100B. (Each 62-bit wide pod is supplied with an "MT600" ROM, but only one ROM need be installed to support any of the 62-bit pods.)

8086/8088 Disassembly ROM: The "M8600" disassembly ROM (supplied with the 16I-086 pod) must be installed on a DI-012 ROM Disassembly Board inside the ML4100B if the user desires to view data on the Disassembly screen.

2. OVERVIEW OF 16-BIT MICROPROCESSOR POD OPERATION

The 16I-086 Microprocessor Pod clips directly over an 8086 or 8088 DIP. The pod collects data (at up to 62 channels) at microprocessor clock speeds of up to 10 MHz, and formats it for logic analysis and full program disassembly (including all exceptions and bus cycles) by the ML4100B.

The 8086 and 8088 microprocessors have two operating modes, Minimum and Maximum. In Minimum mode, the processors provide few indications of their internal operation; in particular, no indication of instruction queue activity is available outside the processor. For Minimum-mode processors, the 16I-086 Microprocessor Pod has only one mode of operation--Transparent.

In Maximum mode, these processors provide queue status information for use by coprocessors, and the 16I-086 pod provides three operating modes of its own: Transparent, Filtered, and Jump. The Filtered and Jump modes use the queue status information to record only specific processor cycles.

There are a total of eight combinations of processor modes (Minimum and Maximum) with pod modes (Transparent, Filtered, and Jump), and the pod produces disassembled code in each of them. In the Transparent mode, all bus cycles are passed along to the ML4100. In Filtered and Jump modes, special hardware in the pod tracks every bus cycle of the microprocessor and determines if the particular data will be recorded in the pod's current operating mode. In the Filtered mode, all unused prefetch cycles are removed in the pod hardware so they won't activate unwanted trigger events and won't be stored in the ML4100 memory. In the Jump mode, all bus cycles are filtered out except for changes in program flow (jumps, subroutine calls, interrupts, etc.), resulting in a disassembled "map" of where code was actually executed.

The ML4100 Disassembly display screen details all microprocessor data cycles "unscrambled" and put back together with their corresponding instructions. The State display screen shows data in the order in which it was transmitted by the pod. Data may also be viewed on the Timing display screen; however, there is no internal clocking capability, and only in the Transparent or Jump modes are samples stored in increasing time order. (Filtered mode delays storage of instruction cycles relative to that of data cycles.)

B. HARDWARE

1. SWITCH SETTINGS

On one end of the 16I-086 pod, there is an access hole for a 4-position DIP switch. This access hole is on the left end of the pod when viewed with the label lettering upright.

Switch 1 selects the processor mode:

Down = MAX (Maximum)

Up = MIN (Minimum)

NOTE: ML4100 power must be cycled (turned off, then on again) whenever the position of this switch has been changed.

Switch 2 selects the data bus width (processor type):

Down = 8 bits (8088 processor)

Up = 16 bits (8086 processor)

NOTE: ML4100 power must be cycled (turned off, then on again) whenever the position of this switch has been changed.

Switch 3 controls instruction byte repacking, and applies to the use of the Filtered pod mode on 16-bit processors (8086F) only:

Down = repack instruction bytes into words before recording (unless a data cycle occurs on the processor bus after the low instruction byte is used and before the high instruction byte is used).

Up = send instruction bytes to the ML4100 as soon as they are used; make no specific attempt to repack the bytes. (Repacking may occur occasionally.)

NOTE: Switch 3 normally must be Down, or the disassembler may be unable to disassemble portions of the recorded data. This switch may be changed during use; i.e., the ML4100 power need not be cycled after resetting the switch. (See also Sections E.2 and G.3.)

Switch 4 controls Jump mode operation (for Maximum-mode processors only):

Down = Jump mode records only program flow discontinuities due to the INTR interrupt.

Up = Jump mode records program flow discontinuities due to all causes (jumps, calls, returns, software and hardware interrupts, and instruction traps).

NOTE: This switch may be changed during use; i.e, the ML4100 power need not be cycled after resetting the switch. The setting of this switch should be examined (and changed if necessary) whenever the user chooses to operate the pod in Jump mode. (See also Section F.)

2. SIGNAL ASSIGNMENTS

The signal assignments vary with the pod mode, which is dependent upon processor type (8086 or 8088) and processor mode (Minimum or Maximum). The following table lists the 8 combinations of processor modes and pod modes, and shows how they are designated on the ML4100 display screens.

Figure 37
DESIGNATIONS OF 16I-086 PROCESSOR/POD MODES

Modes Desig.	Processor Mode	Data Bus Width	Pod Recording Mode
8086mn	Minimum	16 bits	Transparent
8086T	Maximum	16 bits	Transparent
8086F	Maximum	16 bits	Filtered
8086J	Maximum	16 bits	Jump
8088mn	Minimum	8 bits	Transparent
8088T	Maximum	8 bits	Transparent
8088F	Maximum	8 bits	Filtered
8088J	Maximum	8 bits	Jump

Signal class assignments for the 62 available probe lines (62-32, 30-0) are:

Figure 38
SIGNAL CLASS ASSIGNMENTS--OVERVIEW

	ALL MODES:	
Binary:	Probe 18 = Side Probe 1 Probe 17 = Side Probe 0 Probe 16 = Reset Flag	
	BOTH MINIMUM PROCESSOR MODES: (8086mn & 8088mn)	ALL MAXIMUM PROCESSOR MODES: (8086T, 8086F, 8086J, 8088T, 8088F, 8088J)
Status:	Probes 60-52	Probes 62-52
	ALL MODES:	
Address:	Probes 51-32 (32 = LSB)	
	ALL 8086 MODES: (8086mn, 8086T, 8086F, 8086J)	ALL 8088 MODES: (8088mn, 8088T, 8088F, 8088J)
Data:	Probes 15-0 (0 = LSB)	Probes 7-0 (0 = LSB)

Figure 39
STATUS SIGNAL ASSIGNMENTS

(Asterisk denotes active low)

Probe No.	Position Label on Trigger Screen	8086mn	8086T	8086F	8086J
52	B0	BHE*	BHE*	BHE*	BHE*
53	B1	PHLD	LOCK*	LOCK*	LOCK*
54	B2	IAK*	S0*	S0*	S0*
55	B3	DT/R*	S1*	S1*	S1*
56	B4	M/IO*	S2*	S2*	S2*
57	B5	S3	S3	S3	S3
58	B6	S4	S4	S4	S4
59	B7	S5	S5	S5	S5
60	B8	S6(=0)	S6	S6	S6
61	B9	--	(always 1)	FL*	ID*
62	B10	--	(always 1)	FH*	SRC

Probe No.	Position Label on Trigger Screen	8088mn	8088T	8088F	8088J
52	B0	BHE*	BHE*(=1)	BHE*(=1)	BHE*(=1)
53	B1	PHLD	LOCK*	LOCK*	LOCK*
54	B2	SS0*	S0*	S0*	S0*
55	B3	DT/R*	S1*	S1*	S1*
56	B4	IO/M*	S2*	S2*	S2*
57	B5	S3	S3	S3	S3
58	B6	S4	S4	S4	S4
59	B7	S5	S5	S5	S5
60	B8	S6(=0)	S6	S6	S6
61	B9	--	(always 1)	FL*	ID*
62	B10	--	(always 1)	(always 1)	SRC

Most status signals are sampled directly at the microprocessor, and their names correspond directly to 8086 or 8088 signals. Those status signals whose names do not correspond to 8086 or 8088 signals are below.

Figure 40
STATUS SIGNALS WHOSE NAMES DO NOT CORRESPOND
TO 8086/8088 SIGNALS

PHLD is high (1) for the first recorded microprocessor-caused bus cycle following an occurrence of HLDA. This signal is present in Minimum mode only, and indicates that a DMA cycle (or cycles) probably occurred just prior to this microprocessor cycle. DMA cycles are not recorded in trace memory for Minimum-mode processors.

IAK* is low (0) for INTA* cycles, and is an approximate substitute for S0*, which is not available on the Minimum-mode 8086.

FL* is low (0) if the lower data byte is the first byte of an instruction.

FH* is low (0) if the upper data byte is the first byte of an instruction.

SRC is high (1), for 8086J and 8088J modes, if the indicated byte on the data bus is the Source of a discontinuity, and low (0) if the byte is the Destination of a discontinuity.

ID* is low (0), for 8086J and 8088J modes, if this Destination entry in trace memory is due to a discontinuity caused by an INTR hardware interrupt.

Figure 41
BINARY FIELD SIGNAL ASSIGNMENTS

Probe No.	Signal	Function
18	Side Probe 1	Sampled about 12 ns after the processor data bus is sampled (at the end of each bus cycle)
17	Side Probe 0	Sampled about 12 ns after the processor data bus is sampled (at the end of each bus cycle)
16	Reset Flag	A latched signal set on the first bus cycle following reset (fetch at 0FFFF0H)

3. STATUS CODES

The STS field (on the Trigger setup screen and the State display screen) specifies status information about processor bus cycles. The signals on Probes 56-54 (usually processor signals S2*, S1*, and S0*) specify the purpose of the cycle (I/O read, I/O write, memory read, memory write, fetch, or interrupt acknowledge). The following table shows status codes for the various types of cycles.

Figure 42
TYPICAL STATUS CODES
FOR TRIGGERING ON SPECIFIC CYCLE TYPES

	ALL MAX. MODE PROCESSORS	8088mn	8086mn
Interrupt Acknowledge	XXXXXX000XX	XXXX000XX	XXXX000XX
I/O Read	XXXXXX001XX	XXXX001XX	XXXX001XX
I/O Write	XXXXXX010XX	XXXX010XX	XXXX011XX
Fetch (except 8086mn)	XXXXXX100XX	XXXX100XX	
Fetch (8086mn)			XXXX101XX
Memory Read (Fetch in 8086mn)	XXXXXX101XX	XXXX101XX	
Memory Write	XXXXXX110XX	XXXX110XX	XXXX111XX

The signals on Probes 58 and 57 (S4 and S3) indicate which segment of memory was accessed by the bus cycle. The following examples specify memory segments in addition to cycle type.

Figure 43
TYPICAL STATUS CODES
FOR TRIGGERING ON SPECIFIC MEMORY SEGMENT ACCESSES

	ALL MAX. MODE PROCESSORS	8088mn	8086mn
Read from ES:	XXXX00101XX	XX00101XX	XX00101XX
Read from SS:	XXXX01101XX	XX01101XX	XX01101XX
Read from CS:	XXXX10101XX	XX10101XX	XX10101XX
Read from DS:	XXXX11101XX	XX11101XX	XX11101XX
Write to ES:	XXXX00110XX	XX00110XX	XX00111XX
Write to SS:	XXXX01110XX	XX01110XX	XX01111XX
Write to CS:	XXXX10110XX	XX10110XX	XX10111XX
Write to DS:	XXXX11110XX	XX11110XX	XX11111XX

By Intel convention, the signals on Probes 60 and 59 (S6 and S5) indicate the following:

If S6 = 0, then the microprocessor caused this bus cycle.

If S5 = 0, then INTR interrupts are disabled.
If S5 = 1, then INTR is enabled.

For Maximum-mode processors only:

If S6 = 1, then a coprocessor or other DMA device caused this bus cycle.

If S5 = 0, then the coprocessor is an 8087.
If S5 = 1, then the coprocessor is an 8089.

Note than when disassembly of 8087 instructions is enabled (see Section H.2.g, "X Key"), or when DMA cycles are present, the disassembler relies upon the above conventions to distinguish 8086/8088 cycles from all others.

4. POD CONNECTION ERROR WARNING

A "Pod Connection Error" warning is given by any of the ML4100 microprocessor pods if the pod is not properly connected to the target microprocessor, or if power is not applied to the target microprocessor. The 16I-086 pod is no exception, but this pod also issues the warning for another reason.

If the DIP switch settings indicate that the microprocessor is running in Minimum mode, but the MIN/MAX pin of the target microprocessor indicates Maximum mode (or vice versa), a "Pod Connection Error" will be issued. To correct this condition, the DIP switch settings must be changed and ML4100 power must be cycled.

The 8087 and 8089 coprocessors do not have a MIN/MAX pin, and the user may defeat the warning hardware to allow the pod to be clipped directly to one of these coprocessors. To defeat the warning, open the pod box and find JP2 (next to U42), then move the shorting jumper to short JP2 Pin 2 (middle pin) to JP2 Pin 3 (toward U49). (The shorting jumper is normally installed from JP2 Pin 1 to JP2 Pin 2.)

C. MINIMUM PROCESSOR MODE (TRANSPARENT POD MODE ONLY) (8086mn and 8088mn)

The 8086 and 8088 microprocessors operate in two modes, Minimum and Maximum. Processors operating in Minimum mode provide few indications of their internal operation; in particular, there is no indication of instruction queue activity outside the processor. For Minimum-mode processors, then, the ML4100B offers only the Transparent pod mode, which causes the 8I-086 pod to transmit processor bus cycles to the ML4100B trace memory in the same order in which they occur.

Overfetched instructions (those instructions prefetched but later discarded due to a jump, interrupt or other discontinuity in program flow) may trigger the analyzer in spite of the fact that these instructions were not executed. Triggering on instructions which are occasionally overfetched can usually be overcome by triggering on a later instruction in the same code path (one which is further downstream from a jump, call, return, etc., which is causing the overfetch), or by using a more complicated trigger sequence to determine when the instruction is actually executed.

D. MAXIMUM PROCESSOR MODE WITH TRANSPARENT POD MOD (8086T and 8088T)

Because processors operating in Maximum mode provide queue status information, the 16I-086 microprocessor pod offers three "POD MODE" entries on the Format setup screen when the Maximum processor mode has been selected (via the pod DIP switch). The first of the three modes, Transparent (8086T or 8088T), does not use the queue status information; this pod mode causes the pod to send processor bus cycles as they occur. Overfetches are not stripped, and triggerring on overfetches may occur. If the overfetches cause triggering problems, put the pod in Filtered mode.

E. MAXIMUM PROCESSOR MODE WITH FILTERED POD MODE (8086F and 8088F)

1. 8086F AND 8088F MODES

In the Filtered modes (8086F or 8088F), the 8I-086 pod maintains an instruction queue of its own, and transmits the instruction byte from the queue only when the queue status pins on the processor indicate that these bytes are actually used. Overfetched bytes are never sent to the ML4100, and they cannot cause triggering. Instruction bytes are delayed in the pod until their actual execution; they will appear later in the trace buffer relative to nearby data (noninstruction) bus cycles. This slight delay of the instruction bytes has no effect upon the display of trace memory as disassembled code. The delay, and the absence of overfetch, relative to Transparent mode, can be seen only in the State data display.

In Filtered mode, the instruction queue is handled on a byte-by-byte basis. For the 8086, which prefetches words (two bytes) at a time, the pod has the capabiity of repacking the executed instruction bytes into the original words, and (via a DIP switch on the pod) the user can somewhat control the repacking of program bytes. The remainder of this section discusses the action of this switch, and applies only to the 8086F mode.

2. 8086F MODE; REPACKING INSTRUCTION BYTES (See also Section G.3)

In the 8086F mode only, Switch 3 in the Down position causes the pod to delay sending a lower byte of an instruction word until the disposition of the upper byte is known, or until a noninstruction data cycle occurs. (The pod will never allow a noninstruction bus cycle to enter the trace memory ahead of an executed instruction byte which may have caused the data cycle.) So, whenever possible, the pod will repack a low-byte/high-byte pair of instruction bytes when Switch 3 is Down. This repacking will not occur if one of the bytes is not executed, or if a noninstruction bus cycle occurs between the execution of the two bytes. Keeping Switch 3 down uses the available trace memory much more efficiently.

In the Up position, Switch 3 commands the pod, in Filtered mode, to transmit executed instruction bytes from its queue as soon as they are available. Bytes will be repacked into words only if, due to some delay such as the transmission of data cycles, the pod finds that a low/high pair of instruction bytes has been executed and neither has been transmitted to the ML4100B. Thus, with Switch 3 up, repacking will occur only as required; the pod makes no special effort to repack the instruction bytes. The Up position is useful when the processor "hangs" after executing a WAIT or HALT. The pod will empty itself of all executed instructions rather than keep the last byte (if it was a low byte), waiting for another instruction.

Switch 3 may be changed at any time, even while the pod is gathering data; power does not need to be cycled, and the change of pod operation takes place immediately. With Switch 3 in the Up position, trace memory is less efficiently used, and the disassembly software may be unable to disassemble portions of the recorded data. In general, Switch 3 should be Down.

In the 8086F mode, when an instruction byte is transmitted separately from a companion byte in the same prefetched word, BHE* and A0 are modified to refer to the proper byte of the split word.

F. MAXIMUM PROCESSOR MODE WITH JUMP POD MODE (8086J and 8088J)

In the 8086J and 8088J modes, the ML4100 records discontinuities in program flow due to jumps, calls, returns, interrupts, etc. At each discontinuity, two entries are recorded in trace memory: the Source and the Destination.

The Source is the address and the first byte of the instruction which caused the discontinuity, or was last executed before a hardware interrupt. The Destination is the address and first byte of the instruction which was executed just after the discontinuity. A special status bit is high for the Source entry and high for the Destination entry. Another status bit is low for those Destination entries which are caused by INTR hardware interrupts.

Switch 4 of the DIP switch allows the user to select between recording only discontinuities due to the INTR hardware interrupt (Down) or all discontinuities (Up).

The Jump mode is especially useful for debugging runaway software or following the decision-making logic in a block of code. In this mode, only the first bytes of the instructions on each side of a discontinuity are available to the disassembler; no other bytes of a given instruction are available. Therefore, only the mnemonic can be displayed by the disassembler, if that much; no operands or data cycles are available for display.

For some instructions, the first byte alone does not specify a mnemonic. For these instructions, a guess (indicated by a question mark) is made at the mnemonic, and only the displayed byte value and address are certain. By using the address value, the user can locate the actual instruction in an assembled code listing.

G. TRIGGERING AND DATA QUALIFICATION

1. TRIGGERING SEQUENCE LIMITATIONS IN 62-BIT OPERATION

The basic width of the ML4100B data path is 32 bits. In order to accommodate 16-bit processors, the ML4100B divides a processor cycle into halves, and these halves are stored sequentially. (One bit of each set of 32 bits marks which half is which, leaving 31 bits per half and a total of 62 bits to store information.) The basic triggering mechanism in the ML4100B is also 32 bits wide and has up to 13 states. When only the upper 31 bits of information (the Status and Address fields) are used, all triggering capabilities are available (as described in Sections III.E, III.F, and III.G). However, it is often necessary to use the Binary and Data fields to trigger. Special software allows the user to specify a single 62-bit trigger word.

In general, the software allows the user to be unconcerned with the triggering width of the machine; but there are some limitations. Some complex sequences require too many states to be implemented in the 13-state sequence machine. E.g., when the trigger words have something other than Xs in their lower 31 bits, the Trigger Sequence "A + B + C + D" requires 16 states to represent the condition past the first half of the trigger word. This sequence is unavailable when Binary and Data fields (other than "Don't Cares") are specified.

For a User-Defined Sequence, the number of states needed for a given condition level can be calculated using these rules:

> A single word requires two states. If further words are "ORed" in, double the number of states for each additional "true" word (e.g., A as opposed to A*). If words are "ANDed" in, double for each additional "not" word (e.g., A*).
>
> E.g., the number of states required for a given condition level are:

Condition	No. of States
A	2
A + B	4
A + B + C*	4
A & B	2
A* & B*	4
A + B + C	8
A* & B* & C*	8

> Each level consists of two conditions; add the number of states required for each set of two conditions, and multiply the resultant sum by the count for that level; sum at each level to obtain the total number of states. The maximum number of states allowed is 13.

However, it is usually not worth the time to determine the number of states required by a particular sequence. Simply enter the sequence on the Trigger Sequence setup screen, and the ML4100B 62-bit trigger and sequence software will decide if the sequence is too complex. If it is, the ML4100B will display an error message ("Trigger Too Complex") on the Status screen; just remove a word from a condition or reduce the count at a level, and try again.

2. DATA QUALIFICATION IN 62-BIT OPERATION

Due to the 32-bit nature of the ML4100B capture and trigger circuitry, only the upper 31 bits (the default Status and Address fields) of a processor cycle may be used to qualify data storage. Thus cycles may be qualified based on Address and access type, but not on the Data field. Although the qualification decision is based on the first (upper) half of the cycle, the decision holds for the second (lower) half of the cycle; i.e., whole 62-bit frames are either qualified or not qualified.

3. TRIGGERING METHODS WITH PREFETCHING OR BYTE REPACKING

The prefetch mechanism of the 8086 or 8088 causes the microprocessor to fetch some bytes from code space which are subsequently discarded due to a jump, call or other discontinuity in program flow. These fetched but unexecuted instruction bytes are called overfetches, and they present potential problems to those trying to debug code. Typically, the user wants to trigger logic analyzers on executed instructions, and overfetched instructions can cause false triggering when they are included in the triggering setup.

The Filtered pod mode strips overfetched code from the samples sent to the ML4100B, but the Filtered mode cannot be used for all applications (e.g., Minimum-mode processors), and the 8086F mode presents some triggering complications due to instruction byte repacking. This section suggests methods to overcome triggering problems created by overfetching in the Transparent pod modes, or by repacking in the 8086F mode.

a. Overfetch Problems in the Transparent Mode

For Minimum-mode processors, or for Maximum-mode processors being recorded using a Transparent pod mode, overfetches will sometimes present a problem while trying to trigger the logic analyzer. A few tricks may be used to trigger the analyzer when the instruction is actually executed rather than overfetched.

Method One: (Maximum-mode only) Switch to a Filtered pod mode.

Method Two: If possible, set the trigger word to an instruction downstream from the overfetched instruction, but in the same execution path. To avoid false triggering, this downstream instruction must be far enough away from the cause of the overfetching (a jump, call, return, software interrupt, or other interrupting instruction) that the overfetching does not reach the downstream instruction.

Method Three: Trigger on a noninstruction data access unique to the instruction desired, or another instruction in the same code path. Unexecuted instructions do not perform any noninstruction data accesses.

Method Four: Trigger on a sequence of instruction fetches and/or data accesses unique to the code path desired.

b. **Triggering Problems Due to Instruction Byte Separation within a Word** (8086 Processor only)

The processor object code comprises instructions quantized in units of bytes; that is, the smallest instruction is one byte long, and instructions range in size up to six bytes, with all sizes in between included. The 8086 usually prefetches words; therefore, in general, one prefetch bus cycle may contain bytes belonging to two different instructions. Even if the bytes belong to one instruction, the processor can be viewed as "using" the bytes at different times, not simultaneously. If the 8086 prefetches a single byte or if (in the 8086F mode) the pod sends the bytes of a prefetched word individually, the user may have trouble triggering on a particular instruction byte.

The 8086 will prefetch a single byte only when that byte is a high byte (i.e., has an odd address), and when that byte is the destination of a jump, return or other discontinuity (including return from an interrupt service routine which executed between the low and high bytes of a word). If this odd byte is always accessed singly, the user can simply trigger at the odd address with BHE* set to 0. If, however, this byte may also be prefetched as part of a word at an even address, Method Two or Method Three (below) must be used.

In the 8086F mode, the 16I-086 pod can repack the individual instruction bytes of prefetched words into words. The user can somewhat control the degree to which this repacking occurs via Switch 3 on the pod. Unfortunately, the settings of this switch cannot guarantee that repacking will never occur or always occur. Because the pod alters bits A0 and BHE* according to which byte or bytes are sent to the ML4100 in a given instruction record, and since the user cannot absolutely predict that a given pair of bytes will or will not be repacked, triggering on the address of a particular instruction may be difficult.

The following methods are recommended to overcome the problems created by prefetching and repacking uncertainties:

Method One (for triggering on a low byte): Specify the address in the hexadecimal Address field, and place an "X" in the BHE* status bit.

Method Two (for triggering on a high byte): On the Format setup screen, create a binary Address field duplicating the four least-significant address bits (A3-A0, Probes 35-32), then specify the main portion of the address in the hex field, place "0" in the BHE* Status bit of the trigger word, and place an "X" in A0 of the binary Address field. Note that placing an "X" in A0 will cause a "?" to appear in the least significant nibble of the hexadecimal Address field; the values of A3-A1 can be determined by examining the binary Address field. When creating the new field on the Format setup screen, note that the "Sense" of the field should be "Inverted" (INV).

Method Three (for triggering on a high byte): Using two trigger words (for example, A and B), enter the even address of the word containing the high byte in A, and the odd address of the byte itself in B. Place "0" in BHE* of both words, and trigger on "A or B" or similar sequence.

H. DISPLAY

1. GENERAL

With the 16I-086 pod in Transparent mode, the State display shows all instruction fetches and data cycles in the same order in which they occurred on the bus. In Filtered mode, the order of cycles in the trace buffer will not match the actual order of cycles on the processor bus (see Section E, "Filtered Mode"), and any unused fetches (fetches following an instruction causing a program discontinuity) are disqualified by the pod and not stored in the trace buffer.

The Disassembly display groups these program fetches and data cycles into assembly language instructions. Regardless of whether or not the pod is filtering unused fetches, these unused fetches are not displayed on the Disassembly screen. The current pod mode is shown in a reversed field at the top of the Disassembly screen.

The formatting and display of data is described generally for the ML4100 in Section IV, "Operation." Further details of data display when using 16-bit microprocessor pods, including typical Disassembly display screens, keyboard functions, and display formats, follow.

Figure 44
DISASSEMBLY DISPLAY
(TRANSPARENT AND FILTERED MODES)

```
STATE  ADDR   DISASSEMBLY    X: 8086T+8087
 TRIG EXTERNAL INTERRUPT TYPE FF
      CS:003FC>063A(IP)  CS:003FE>0000(CS)
      SS:20657<F202(F)   SS:20655<0000(CS)
      SS:20653<0632(IP)
00008 0063A   WAIT
00008 0063B   NOT    AX
00011 0063D   TEST   AL,[0632]
      DS:30632>FA
00013 00641   JNE    00644          (short)
00017 00644   MOV    [0632],AL
      DS:30632<FE
00018 00647   STC
00019 00648   ADC    AX,AX
00021 0064A   ADC    AX,AX
f1=Pos. Cursor f2=Spd: slow f3=Cycles: on
```

Figure 45
DISASSEMBLY DISPLAY
(JUMP MODE)

```
STATE  ADDR   DISASSEMBLY            8086 J
 TRIG 00600   MOV    opcode=BB  (DEST)
00001 00615   LOOP   opcode=E2  (SRC)
00002 00609   MOV    opcode=8A  (DEST)
00003 00615   LOOP   opcode=E2  (SRC
00004 00609   MOV    opcode=8A  (DEST)
00005 00615   LOOP   opcode=E2  (SRC
00006 00609   MOV    opcode=8A  (DEST)
00007 00615   LOOP   opcode=E2  (SRC
00008 00609   MOV    opcode=8A  (DEST)
00009 0061D   CALL?  opcode=FF  (SRC
00010 00663   ESC    opcode=DD  (DEST)
00011 00666   RET    opcode=C3  (SRC
00012 0061F   MOV    opcode=BC  (DEST)
00013 00622   CALL   opcode=9A  (SRC)
f1=Pos. Cursor f2=Spd: slow f3=Cycles: on
```

2. KEY FUNCTIONS ON THE DISASSEMBLY DISPLAY SCREEN

a. f1 Key (Position Cursor)

To move directly to any location in the buffer, depress the f1 key, enter the state number you wish to display, and depress f1 again. Negative state numbers correspond to cycles before the trigger event; positive state numbers correspond to cycles after the trigger. (If f1 is inadvertently depressed, depress it again to leave the cursor at its current location.)

b. Up/Down Arrow Keys (Scroll through Buffer)

To move through the buffer backward or forward, use the Up or Down Arrow keys.

c. f2 Key (Scroll Speed)

To scroll slowly through the buffer (one instruction at a time), use the f2 key to set the scroll speed to Slow. To scroll faster, depress the f2 key to toggle the scroll speed to Fast.

d. f3 Key (Data Cycles On/Off)

To view the disassembled instructions without their data cycles, depress the f3 key. This gives a display very similar to an actual code listing and makes it easier to locate a particular section of code. To view the data cycles associated with the instruction, depress the f3 key to toggle the data cycles back on.

e. f4 Key (Print Screen)

To print a hardcopy report of the Disassembly screen, depress the f4 key. (There is no on-screen prompt for this function.) Note that the RS-232C option must be installed; this is included in the ROM Emulator Module, RE-016.

f. DISASM Key (Display Trigger Condition)

To display the trigger event, depress the DISASM key once to enter the Disassembly screen, and again to display the trigger event.

g. X Key (8087 Disassembly--Maximum processor mode only)

To alternately enable or disable the display of "ESC" instructions as 8087 mnemonics, depress the X key.

3. DISASSEMBLY DISPLAY FORMATS

a. Instruction Display

As with 8-bit microprocessor pod disassembly, each instruction displayed contains a state number corresponding to the instruction position relative to the trigger event, the value of the Program Counter when the instruction was fetched (PC), and the mnemonic and operands for the instruction itself.

b. Data Cycles Display (including 8087 data cycles)

Data cycles are displayed on lines following the instruction that caused them. Display of these cycles may be turned on or off by toggling the f3 key.

Each data cycle contains an indication of its segment (ES:, DS:, CS:, SS:, or I/O:), an address, an indication of whether data was read from (>) or written to (<) the address, and the data itself. Each address consists of five hexadecimal characters (20 bits), while data may be a byte (2 characters), or a word (4 characters) for non-8087 instructions. Although the bytes of a word may have been accessed in two separate bus cycles (e.g., an 8086 accessing a word at an odd address or an 8088), the two bytes will be combined into a single word for display.

Some instructions (such as calls, returns, etc.) have some of their data cycles labelled to identify the Instruction Pointer, "(IP)", and CS Register, "(CS)". These labels may appear more than once in the same set of bus cycles, since often the current value is written to the stack and a new value is then read from memory.

The data cycles of repeated string instructions will be displayed, two per line, on up to 13 lines. To view those cycles beyond these 26 bytes or words, the user must search through the State display data.

The data cycles of 8087 instructions may be as long as 94 bytes. These will be shown on multiple lines, as necessary. The bytes of each separate floating point or integer value will be grouped under a separate address. The 14 bytes comprising the internal status of the 8087, when read or written, will be grouped under a single address. The address will be that of the lowest addressed byte, and that low byte will be the rightmost of the group.

If some or all of the data cycles of an 8087 instruction are missing from the trace memory (possibly because they occurred much further downstream), their place will be denoted by "XX" in the data.

The current version of the 8087 will access only the data segment of memory. If an 8087 instruction is preceded by a segment override prefix, the 8086/8088 will read data from some segment other than DS:; the 8087 then accesses the remaining data bytes in DS:. In this situation, the data cycles of the 8087 instruction will blink to warn of the mixed segment access by the processor-coprocessor pair.

c. Interrupts

Interrupts will be labelled as "EXTERNAL" or "SOFTWARE" (or, in some cases, even more specifically, such as "DIVIDE OVERFLOW", etc.), and their type (vector number) will be displayed. Additionally, the data cycles will be labelled during the stacking of the Instruction Pointer, "(IP)", the CS register, "(CS)", and the Flag register, "(F)". The new Instruction Pointer and CS register retrieved from the vector area will also be labelled.

d. Question Marks

Occasionally (often just before interrupts) when operating the pod in a Transparent mode, some instructions will be labelled with a question mark. In these instances, the disassembler is unable to determine if these instructions actually were executed.

e. "UNABLE TO DISASSEMBLE" Messages

Accurate disassembly of the 8086 or 8088 bus cycles requires extensive pattern matching and decision making. For Transparent pod modes, identifying those code segment bytes which are the first bytes of instructions is a difficult task requiring the disassember to make initial guesses concerning the first bytes of instructions and the location of data cycles relating to a given instruction. These guesses are modified until the trace memory is disassembled without any discrepancies in fitting known patterns to each instruction.

The Filtered modes are simpler to disassemble, due to the presence of "first-byte" status bits on instruction bytes, but locating the noninstruction cycles associated with certain instructions still requires iterative matching of patterns throughout the trace buffer.

The only recording modes which do not require extensive effort to disassemble are the Jump modes.

Occasionally, the trace memory may contain a singularity (an abrupt, small irregularity) through which disassembly cannot pass without a pattern discrepancy. Such a singularity can arise from the use of data qualification, or due to the resetting of the microprocessor while recording. The disassembler will recover from these singularities in several ways, depending mostly upon how large a piece of trace memory was successfully pattern-matched just preceding the singularity, and whether or not there are signs that the processor was reset.

If a reset occurred, the disassembler will display a line with "RESET" in the Address field, and then the pattern matching will resume past the singularity. If the singularity is not due to reset, and a significant number of instructions matched their specific patterns before the singularity, the dissassembler will display "UNABLE TO DISASSEMBLE XXX CYCLES," where XXX is the number of confusing cycles. Disassembly will proceed beyond the singularity.

If too few instructions are successfully matched before a singularity disrupts the disassembly, the disassembler will discontinue its effort to completely disassemble the trace buffer, and will display a message "CAN'T DISASSEMBLE--CS READS USED AS OPCODES." From that message forward, the mnemonic that each code space read would represent if it were the first byte of an instruction is displayed. These mnemonics are preceded by a question mark to indicate that the disassembler could make no sense of the recorded data.

XII. SERIAL DATA ANALYSIS POD (8RS-232)

(blank)

A. INTRODUCTION

1. OVERVIEW

The 8RS-232 Serial Data Analysis Pod enables the ML4100 to monitor and analyze all transfers occurring over an EIA RS-232C/V.24 serial interface. Up to 1000 synchronous or asynchronous events, during either full-duplex or half-duplex channel operations, can be collected. All bus activity, including data transfers, control bits, and framing, parity, and overrun errors, can be displayed for analysis. Data and triggering formats are binary, octal, decimal, or hexadecimal; data may also be displayed in ASCII.

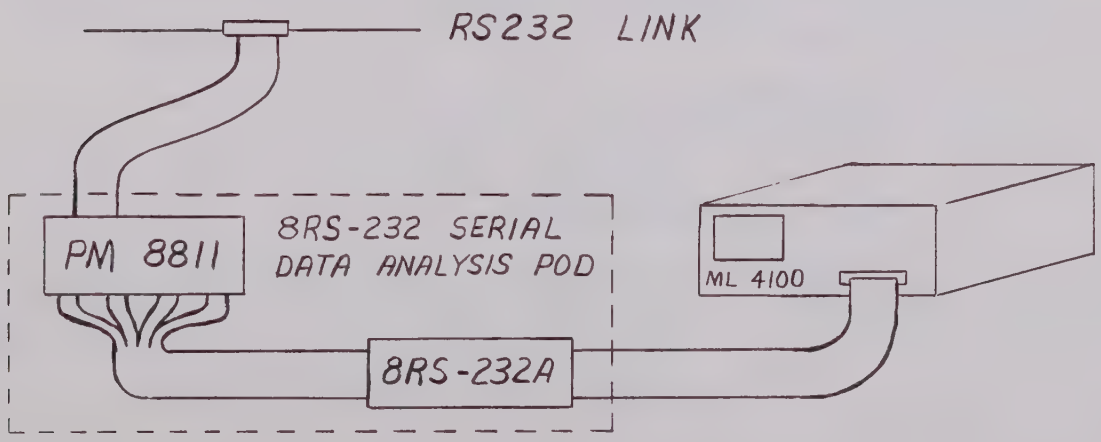
Captured information can be displayed in timing, state, disassembly or text mode. Data may be viewed as received, transmitted, or both in combination; data word lengths may be 5, 6, 7, or 8 bits.

All of the ML4100's standard operating features (such as settable delays, immediate data capture upon powerup, four trigger words, data qualification, and customized formatting of displayed data) are present when using the 8RS-232. In addition, the 8RS-232 Serial Data Analysis Pod provides a special bit when using the internal clock. This bit is used to synchronize on complete character transmissions. Any RS-232C/V.24 data, control, or protocol activity can be used as a trigger event.

2. CONNECTION (See Figure 46)

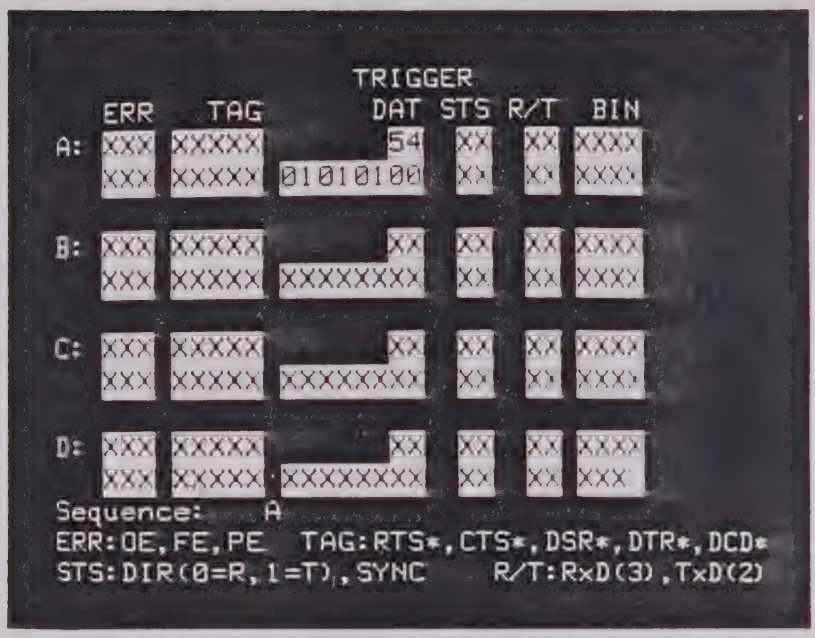
The 8RS-232 pod consists of a Philips PM8811A and an Arium 8RS-232A. This Arium pod connects to the front of the ML4100 in the usual way, and the other side has four cables which attach to the matching labled ports on the Philips pod. The external RS-232 link is connected via a "T" connector and adapter cable. This connector places the pod physically in series with other devices on the channel, but electrically in parallel with them. This allows all monitoring and analysis of the communications interface to be completely transparent to the channel's operation.

Figure 46
8RS-2332 POD CONNECTION



B. TRIGGERING

Figure 47
TRIGGER SETUP SCREEN



1. TRIGGER SETUP SCREEN (See Figure 47)

The Trigger setup screen is used to set up any of the four trigger words (A, B, C, or D). Each word contains six fields: ERR (Errors), TAG (RS-232C protocol signals), DAT (Data), STS (Status), R/T (Receive/Transmit), and BIN (Binary). Any combination of these four trigger words and six fields may be used to designate event collection.

The bit assignments for each of the six fields are displayed on the screen's two bottom lines. The default state field definitions follow; as with any ML4100C setup, these may all be changed by the user via the Display Format setup screen (Section III.H).

a. ERR (Errors)

Bit 2: OE (Overrun Error)--Two serially transmitted words have occurred back-to-back, without any stop bits between them.

Bit 1: FE (Frame Error)--Either start- or stop-bit timing has been violated.

Bit 0: PE (Parity Error)--Parity bit does not agree with the bit computed per switch setting.

b. TAG (RS-232C Protocol Signals)--Handshake protocol

Bit 4: RTS (Receive-to-Send)

Bit 3: CTS (Clear-to-Send)

Bit 2: DSR (Data Set Ready)

Bit 1: DTR (Data Terminal Ready)

Bit 0: DCD (Data Carrier Detect)

c. DAT (Data)

Data word may be 5, 6, 7, or 8 bits in length; the hexadecimal equivalent is entered in the 2-bit field.

d. STS (Status)

Bit 1: Communication Direction
1 = Transmit
2 = Receive

Bit 0: Synchronization (active only in Timing mode)
1 = Current cycle is the first after a new character is detected
0 = All other cycles

When using the internal clock, the analyzer can trigger the sync bit which the 8RS-232 pod creates. This bit is high for one clock cycle after each new character; thus it can be used to trigger on ASCII data sequences even when the analyzer is set to internal clock and is taking many samples over one RS-232 bit time.

e. R/T (Receive/Transmit Pin Selection)

Bit 1: 1 = Receive

Bit 0: 1 = Transmit

Bits 1/0: 11 = Combination Receive and Transmit

f. BIN (Binary)

Bits 3-0: Selection of up to four external lines (probes) for monitoring/analyzing accessory signals

2. SELECTION OF TRIGGER EVENT PARAMETERS (via Trigger setup screen)

To set trigger event parameters, cursor to the appropriate bit within one of the six fields, then enter the desired value via the keypad (X = Don't Care). (See Section III.E.)

C. DISPLAY FORMATS

1. GENERAL

Figure 48 (Display Format setup screen) shows the 8RS-232 pod's default parameters, default probe assignments, and field widths. Note than Lines 5, 11, and 12 do not have assigned probes and that only nine timing lines are displayed under default conditions. In the State mode, all fields default to a binary base except the Data field, which is set to hex. (See Section III.H for a general discussion of how to use this screen.)

The pod operates in two modes: synchronous data capture (indicated by "RS232" in the pod mode filed at the bottom of this screen), and asynchronous data capture (indicated by "RS232T"). In the synchronous mode (the default), one clock cycle occurs for each character detected; in the asynchronous mode (internal clock selected on the ML4100 Clock setup screen), the sample rate is established by the internal clock. To select synchronous mode of pod operation, depress f1; to select asynchronous mode, depress f2.

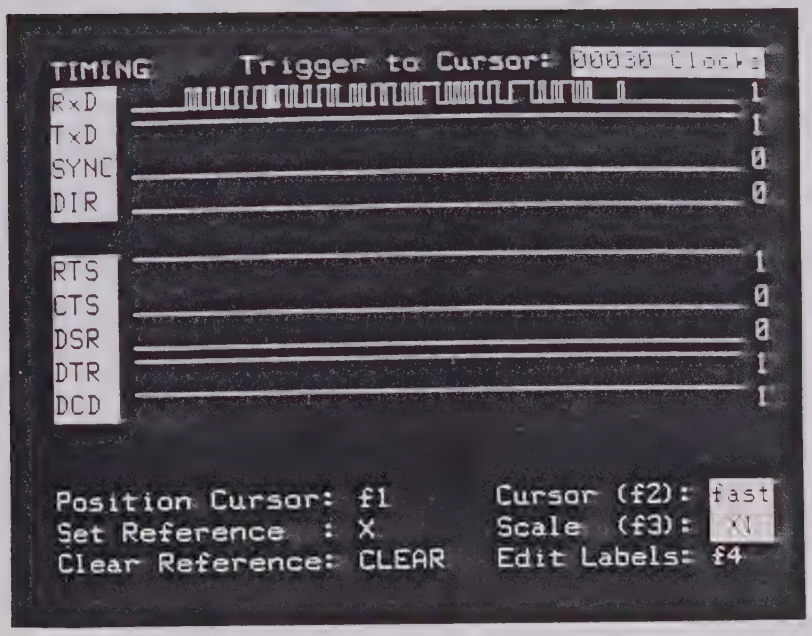
Figure 48
DISPLAY FORMAT SETUP SCREEN

```
                    DISPLAY FORMAT

TIMING:
Line      1  2  3  4  5  6  7  8  9 10 11 12
Probe    01 00 06 07 XX 20 19 18 17 16 XX XX

STATE:       ERR TAG DAT STS R/T
Field          1   2   3   4   5   6   7   8
Base         BIN BIN HEX BIN BIN BIN OFF OFF
High Probe    23  20  15  07  01  05  00  00
 Low Probe    21  16  08  06  00  02  00  00
Data Sense   NON NON NON NON NON NON NON NON
Bases: A=ASC B=BIN C=OCT D=DEC E=HEX F=OFF
Senses: 0=NON (non-inverted), 1=INV

POD MODE (also sets defaults): RS232
f1=RS232   f2=RS232T
```

2. TIMING DISPLAY SCREEN (See Figure 49)

The Timing display screen shows the Receive and Transmit Data lines, synchronization signal (SYNC), and data transfer direction (DIR), as well as the five TAG field lines (communications protocol). (See Sections IV.A and IV.B for a general discussion of how to use this screen.)

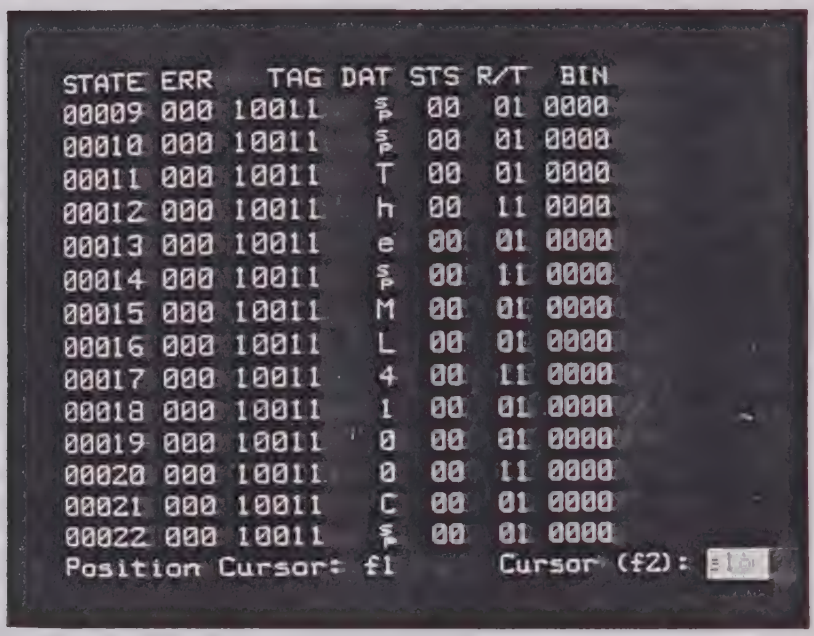
Figure 49
TIMING DISPLAY SCREEN



3. STATE DISPLAY SCREEN (See Figure 50)

The State display screen gives reference numbers to each word sampled. The trigger cycle is identified by "TRIG" appearing in place of the reference number. The current memory being displayed (A or B) is shown at the center of the screen's bottom line.

Figure 50
STATE DISPLAY SCREEN



a. ERR

The Errors field depicts overrun, framing, and parity error status in the left, middle, and right bits, respectively.

b. TAG

This field describes (from left to right) RTS (Request-to-Send), CTS (Clear-to-Send), DSR (Data Set Ready), DTR (Data Terminal Ready), and DCD (Data Carrier Detect) status. The DAT column defaults to hexadecimal format.

c. STS

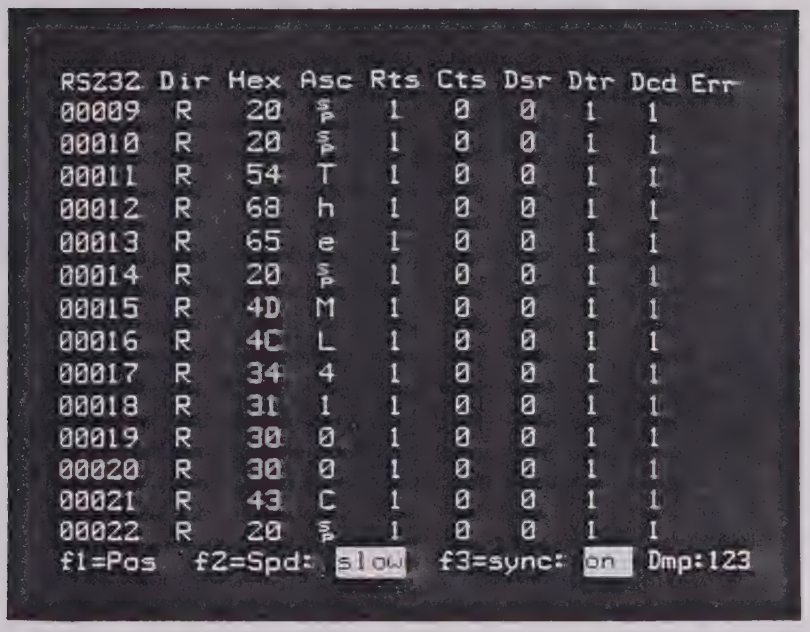
In the Status field, the leftmost bit indicates the direction of the communications flow; a "0" denotes Receive data (Pin 3), and a "1" denotes Transmit data (Pin 2). In the righthand bit, a "1" indicates the first clock cycle after a new character is received in timing mode; a "0" is displayed at all other times.

d. BIN

This field is used for external probes which may be connected to monitor additional signals; the default mode is to Probes 2-5.

4. DISASSEMBLY DISPLAY SCREEN (See Figure 51)

Figure 51
DISASSEMBLY DISPLAY SCREEN



a. RS-232

This leftmost field clearly marks the trigger word and assigns reference numbers to all word samples in relation to that trigger word.

b. DIR

This field indicates the direction of the data flow (Receive or Transmit).

c. HEX and ASC

These fields display the decoded characters in hex and ASCII formats.

d. TAG

This field displays the individual TAG bits.

e. ERR

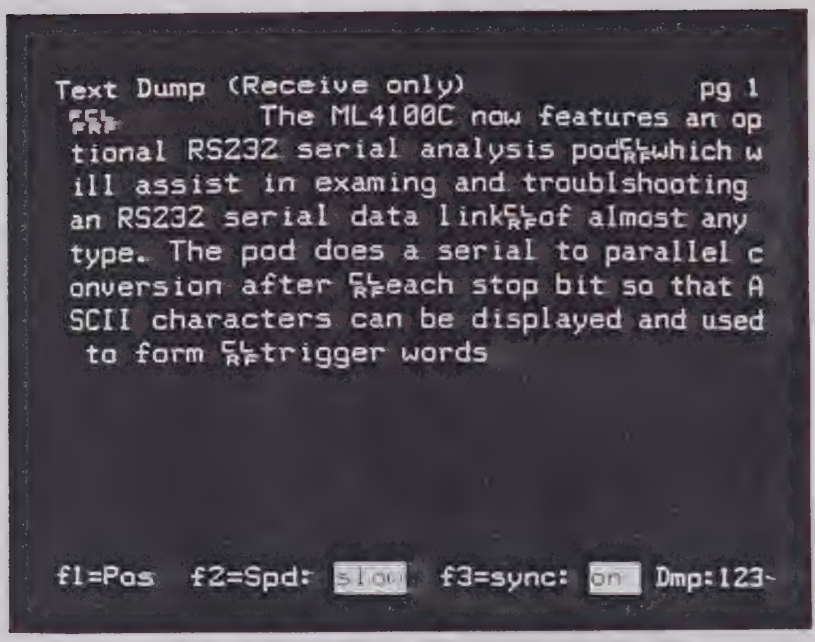
This field displays errors occurring in character detection.

f. SYNC (only in Timing mode)

Depressing the f3 key toggles between displaying all samples (sync = off), and only those for which the sync bit was high (sync = on).

5. TEXT DISPLAY SCREEN (See Figure 52)

Figure 52
TEXT DISPLAY SCREEN

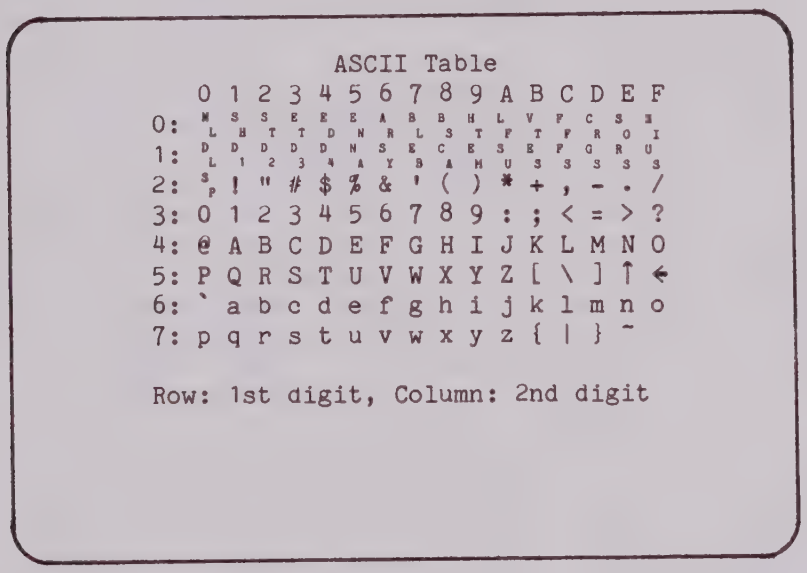

Text Dump (Receive only) pg 1
The ML4100C now features an op
tional RS232 serial analysis pod which w
ill assist in examing and troublshooting
an RS232 serial data link of almost any
type. The pod does a serial to parallel c
onversion after each stop bit so that A
SCII characters can be displayed and used
to form trigger words
f1=Pos f2=Spd: slow f3=sync: on Dmp:123-

This special mode displays data in ASCII(text) format. To enter the Text display mode and get a text dump in ASCII format, first depress DISASM (to access the Disassembly display screen), then select either Transmit data only (by depressing "1"), Receive data only (by depressing "2"), or both Transmit and Receive data simultaneously (by depressing "3").

D. SPECIAL FUNCTIONS (See Figure 53)

A hex-to-ASCII conversion table can be displayed for the user's reference. To access the ASCII conversion table, depress the S. FUNC key (to enter the screen), then depress "E". The most significant hex character indexes the row, and the least significant indexes the column. The ASCII representation is read at the row-column intersection.

Figure 53
HEX-TO-ASCII CONVERSION TABLE


ASCII Table
0 1 2 3 4 5 6 7 8 9 A B C D E F
0:
1:
2: ! " # $ % & ' ( ) * + , - . /
3: 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
4: @ A B C D E F G H I J K L M N O
5: P Q R S T U V W X Y Z [\ ] ↑ ←
6: ` a b c d e f g h i j k l m n o
7: p q r s t u v w x y z { | } ~
Row: 1st digit, Column: 2nd digit

E. RS-232C COMMON PIN ASSIGNMENTS

Type	Pin No.	Description
Ground	1	Protective Ground
	7	Signal Ground (Common Return)
Data	2	Transmitted Data
	3	Received Data
Control	4	Request to Send
	5	Clear to Send
	6	Data Set Ready
	8	Received Line Signal Detector
	20	Data Terminal Ready
	21	Signal Quality Detector
	22	Ring Indicator
	23	Data Signal Rate Selector (DTE/DCE Source)
Timing	15	Transmission Signal Element Timing (DCE Source)
	17	Receiver Signal Element Timing (DCE Source)
	24	Transmit Signal Element Timing (DTE Source)
Secondary Data	14	Secondary Transmitted Data
	16	Secondary Received Data
Secondary Control	9	(Reserved for Data Set Testing)
	10	(Reserved for Data Set Testing)
	11	(Unassigned)
	12	Sec. Received Line Signal Detector
	13	Sec. Clear to Send
	18	(Unassigned)
	19	Secondary Request to Send
	25	(Unassigned)

Most RS-232C applications use only the Ground, Data, and Control pins, and rarely are all of these used together. The Timing pins are used in very-high-speed data transmission. Secondary pins are used only in special applications.

APPENDIX A

SPECIFICATIONS

All specifications are valid with the 32-Channel (Standard) Logic Pod unless otherwise noted.

A. DATA INPUTS

Selectable modes	4, 8, 16, or 32 channels
Input voltage	
Minimum (sensitivity)	+/-600 mV (from threshold)
Maximum (operating)	+/-16 V
Maximum (absolute rating)	+/-20 V
Threshold	TTL or variable
Variable threshold range	+/-9 V
Input impedance	100 kohms, 6 pF

B. CLOCK (Single)

Internal (asynchronous)	
Rate	5Hz - 100 MHz
External (synchronous)	
Data setup time	5 nsec maximum
Data hold time	1 nsec maximum
Clock period	14 nsec minimum
Pulse width	7 nsec minimum
Clock Qualifier (one) (external clock only)	
Timing and voltage	(Same as for DATA INPUTS)

Recording Width	Maximum Clock Rate: Internal:	External:
4 channels	100 MHz	70 MHz
8 channels	50 MHz	50 MHz
16 channels	25 MHz	25 MHz
32 channels	12.5 MHz	12.5 MHz

C. TRIGGERING (Sequential & combinational)

Trigger words/events	Four
Trigger word width	32 bits maximum (consistent with recording width)
	19 predefined and 1 user-defined, as limited by recording width:

At Recording Width:	Available Trigger Sequences:
4 channels	0, 1, 2, 3
8 channels	0, 1, 2, 3
16 channels	0, 1, 4 and user-defined
32 channels	All (including user-defined)

Trigger Delay	0 - 50,000 clocks

D. DATA QUALIFICATION
(32-channel mode only)

Qualification Modes	
State	Occurrence of selected words toggles recording on/off
Combinational	Records only those words which match qualification words

E. MEMORY

At Recording Width:	Recording Depth:
4 channels	8000 samples/channel
8 channels	4000 samples/channel
16 channels	2000 samples/channel
32 channels	1000 samples/channel

F. STATE DISPLAY MODE	Formatted word list (14 recorded words in up to 8 fields)
Selectable number base for each field	ASCII, binary, octal, decimal, hexadecimal
Selectable bit groupings for each field	Up to 32 bits wide
Data is repeatable in different number bases and field groupings	
G. TIMING DISPLAY MODE	Up to 12 timing lines (any may be repeated)
Cursors (two)	Current position and reference line
Magnification (horizontal)	Times ten; centered around cursor
Time Measurement	
Units	Clocks (external clock) or time units (internal clock)
Distance measured	If reference line is set, reference-to-cursor; if no reference line is set, trigger-to-cursor
H. DISASSEMBLY DISPLAY MODE	For address, operand(s), bus activity, instruction mnemonic

I. SCROLLING (in all display modes)	
Features	Continuous (auto repeat); fast (one screen at a time) or slow (one sample at a time); left/right for timing display, up/down for others Absolute cursor positioning (in relation to trigger); instantaneous repositioning at trigger
J. CRT	5"; green phosphor screen
K. PHYSICAL MEASUREMENTS	
Size	12" x 5" x 15" (plus handle and feet)
Weight	22 lbs
L. INPUT POWER	115 VAC, 60 Hz, 200 W
M. ENVIRONMENTAL	
Temperature	0-40 deg C operating
Relative humidity	5-95%, noncondensing
N. ROM EMULATOR MODULE (RE-016) (optional)	(Connects to back panel)
ROMs supported	2716, 2732, 2764, and 27128
ROM socket plugs (4)	
Address bus (all ROMs)	14 lines, assumed common
Data busses (2)	8-bits each; independent; for 8- or 16-bit targets
T/acc	200 nsec
T/ce	200 nsec
T/oe	70 nsec
I/ol	2.1 mA
V/ol	0.5 V
Data interface	RS-232C on back panel; uploads/downloads in hexadecimal format
Data editing	From front-panel keypad

O. 32-CHANNEL LOGIC POD (LP-320)
(included with mainframe) — 4-, 8-, 16-, and 32-channel; operates at full rated speed of mainframe

P. OPTIONAL PODS

4-Channel Logic Pod (LP-040)	100 MHz; glitch capture
Microprocessor Pods	
8I-080	For 8085; 8031/8032; and 8035/8039/8040
8M-080	For 6800/6802/6808
8M-089	For 6809, 6809E
8R-065	For 6502, 6512, 65C02, 65C102, and 65C112
8Z-080	For Z80 and Z80A/B/C
(First pod requires a Disassembly ROM Board, DI-012)	
STD Bus Pod (8P-080)	

APPENDIX B

ASCII (USA STANDARD CODE FOR INFORMATION INTERCHANGE)

ASCII

Bit Number				ROW	COLUMN							
			Bit 6		0	0	0	0	1	1	1	1
			Num- 5		0	0	1	1	0	0	1	1
			ber: 4		0	1	0	1	0	1	0	1
3	2	1	0		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

NUL	Null, or all zeroes
SOH	Start of heading
STX	Start of text
ETX	End of text
EOT	End of transmission
ENQ	Enquiry
ACK	Acknowledge
BEL	Bell, or alarm
BS	Backspace
HT	Horizontal tabulation
LT	Line feed
VT	Vertical tabulation
FF	Form feed
CR	Carriage return
SO	Shift out
SI	Shift in
DLE	Data link escape
DC1	Device control 1
DC2	Device control 2
DC3	Device control 3
DC4	Device control 4
NAK	Negative acknowledge
SYN	Synchronous idle
ETB	End of transmission block
CAN	Cancel
EM	End of medium
SUB	Substitute
ESC	Escape
FS	File separator
GS	Group separator
RS	Record separator
US	Unit separator
SP	Space
DEL	Delete

APPENDIX C

INTEL HEX FORMAT

Intel Hex is an Intel Corporation format which provides a means of storing and transmitting object code. It is composed of variable length records which contain only 7-bit ASCII codes. By convention, usually only long, continuously defined memory images are transmitted, in 16-byte increments and beginning with addresses which are a multiple of 16.

In the actual record, all values are bytes or multiple bytes, and each byte is represented by two ASCII characters (0-9 or A-F). This includes byte count, load address, record type, checksum and data.

The checksum is a two's complement negation of the binary sum of the bytes, starting with the byte count byte and ending with the last data byte.

The format is:

:BCAAAATTHhHhHhHhHhHhHhCC

Where:

: (the ASCII character colon) is the start of the record;

BC is two ASCII characters which represent the hex value of the byte count;

AAAA is four characters representing two bytes of address (most significant byte first);

TT is two ASCII characters which represent the hex value of the record type (00 is a data record, 01 is an end-of-file record, and 02 is an extended-address record for 16-bit microprocessors);

Hh is two ASCII characters which represent the hex value of a data byte; (there may be up to 256 of these, depending on the value of BC above and the device handlers in the equipment; normally, there are 16 or fewer); and

CC is two ASCII characters which represent the hex value of the checksum.

Records are terminated with a carriage return and a line feed.

As an example, the following program would be represented as shown below it:

```
0000     C3 00 00     JMP $

:03000000C300003A
:00000001FF
```

APPENDIX E

ERROR MESSAGES

	Message (Screen)	Condition
1.	1 OR 2 WORDS ONLY (Configuration)	In state qualification only two words are allowed for each entry.
2.	ACCESS DENIED (Setup Directory and B Memory Directory)	The wrong security code was entered while trying to read a protected setup or B Memory file.
3.	ADDRESS NOT FOUND (Search and Compare)	One of the 3 addresses specified in a Compare was not found in either A or B Memory, or the address specified in Fill with Memory (Search) was not found.
4.	ALREADY STARTED (any)	The START key was depressed when the ML4100 was already active.
5.	CANNOT WRITE OVER PROTECTED FILES (B Memory Directory)	An attempt was made to write a B Memory file, with a new configuration differing from the stored configuration, over a protected file; the protected file must be unprotected first.
6.	CHECKSUM ERROR--FILE LOST (Setup Directory, B Memory Directory, and Status)	A checksum error occurred while reading a setup or B Memory file; this indicates a hardware problem on the A/B memory card.
7.	DIRECTORY CHECKSUM ERROR--ALL FILES LOST (Setup Directory, B Memory Directory, and Status)	A checksum error occurred while reading the EEPROM directory information; this indicates a hardware problem on the A/B memory card.
8.	FILE CONFLICTS WITH POD TYPE (Setup Directory)	An attempt was made to read a setup file with the wrong pod connected to the ML4100.
9.	FILE DOES NOT EXIST (Setup Directory and B Memory Directory)	An attempt was made to read an available (non-existent) setup or B Memory file.
10.	FILE NO. INCONSISTENT WITH CONFIGURATION (B Memory Directory)	An attempt was made to save a B Memory file in a file number larger than the new B Memory configuration. (This can occur only when changing to a smaller B configuration.)
11.	FILE WIDTH DIFFERS FROM MACHINE WIDTH (B Memory Directory)	An attempt was made to read a B Memory file stored with a width differing from the current width of the ML4100.

12. FORMAT INCONSISTENT WITH WIDTH (Timing and State)	"Probe" numbers higher than those used in the selected configuration have been used in the display format (see Format screen).
13. ILLEGAL COMBINATION (Status)	The two flashing parameters may not be set to their current values at the same time.
14. IMPROPER FILE ORDER FOR AUTOSEQUENCING (Setup Directory)	An attempt was made to exit the Setup Directory screen with the autosequencing start file number larger than the autosequencing end file number.
15. INCORRECT SEARCH SEQUENCE (Search)	An attempt was made to execute a Search with the first search word after the last search word.
16. LOW PROBE MUST BE LESS THAN HIGH PROBE (Format)	When defining display fields via the Format screen, the "Low Probe" must have a lower probe number than the "High Probe".
17. MEMORY CONFIGURATION INCONSISTENT (B Memory Directory)	An attempt was made to read a B Memory file stored under a configuration differing from the current B Memory configuration.
18. NO DATA IN B MEMORY (B Memory Directory, Search, Compare, Edit, and State)	An attempt was made to save, edit, display, search, or upload B Memory while B Memory is empty.
19. NO DATA RECORDED (any)	The START and STOP keys were depressed but fewer than two qualified clocks occurred after starting but before stopping.
20. NO FIELDS TO EDIT (Edit)	An attempt was made to Edit B Memory with all fields in the Format screen turned off.
21. NO RECORDED DATA (B Memory Directory and State)	An attempt was made to copy A Memory to B Memory when A Memory was empty, or a display key was pressed to view A Memory when it was empty.
22. NO POD INSTALLED (any)	(Self-explanatory.)
23. NOT ACTIVE (any)	The STOP key was depressed when the ML4100 was not active.

24. POD CONNECTION ERROR
(any)

The connection from the "pod" to the circuit under test is either not made or made incorrectly, or the processor is not powered (microprocessor pods). (In the case of the 8086/88 pod, if the MIN/MAX DIP-switch setting on the pod does not agree with the target microprocessor's mode, this error message will appear; to correct the error, change the DIP-switch setting and turn the ML4100 off, then on again.)

25. POD x ROM x NOT USED BUT INSTALLED
(when exiting ROM Emulator editor)

The specified DIP cable is installed in a ROM socket but is not defined on the "ROM Emulator Configuration" Screen.

26. POD x ROM x INSTALLATON ERROR
(when exiting ROM Emulator editor)

The specified DIP cable is not installed correctly in a powered ROM socket but is defined on the ROM Emulator Configuration Screen.

27. SELECTED FORMAT TOO WIDE FOR SCREEN
(Format)

If a display was created from the keyed in parameters one or more of the fields would extend off the right side of the screen.

28. START ADDRESS > END ADDRESS
(Compare)

An attempt was made to execute a Compare with a starting address in memory larger than an ending address.

29. WARNING--UNSAVED NAME & STATUS WILL BE LOST
(Setup Directory and B Memory Directory)

An attempt was made to leave the Setup Directory screen or the B Memory Directory screen without saving a file after changing its name and/or status.

INDEX

(blank)

4-Channel Logic Pod . . . . . I- 4
4-Channel Logic Pod V
4-Channel Logic Pod . . . . . II- 3
4-Channel Logic Pod . . . . . II- 4
4-Channel Logic Pod III-22
4-Channel Logic Pod specifications . . . . . V- 2
32-Channel Logic Pod . . . . . I- 4
32-Channel Logic Pod . . . . . II- 2
32-Channel Logic Pod . . . . . II- 3
32-Channel Logic Pod . . . . . II- 4
32-Channel Logic Pod III-22
32-Channel Logic Pod specifications . . . . . A- 5
A Memory IX
Accessories . . . . . I- 4
ASCII, Hex-to-, conversion table XII-11
ASCII table . . . . . B- 2
Asynchronous operation, STD Bus Pod . . . . . VII- 4
Autosequencing of data Compares . . . . . IX- 7
Autosequencing of data Compares IX-16
B Memory IX
Back panel . . . . . I- 8
Beeps . . . . . I- 9
Brightness adjusting screw . . . . . I- 9
Clock . . . . . IV- 4
Clock, external . . . . . II- 4
Clock, external . . . . . III- 9
Clock, internal . . . . . III- 8
Clock qualification . . . . . VI- 2
Clock qualification . . . . . III- 9
Clock qualifier specifications. . . . . A- 2
Clock rate (speed) III-10
Clock setup screen . . . . . III- 6
Clock setup screen . . . . . III- 6
Clock setup screen . . . . . III- 8
Clock specifications . . . . . A- 2
Color codes of probes . . . . . V- 2
Color codes of probes . . . . . II- 2
Color codes of probes . . . . . II- 4
Color codes of probes III-23
Comparing data (A and B Memories) IX-12
See also Continuous Compare
See also Autosequencing of data Compares
Comparing stored data IX-23
Configuration setup screen . . . . . III- 6
Configuration setup screen III-10
Configuration of B Memory . . . . . IX- 8
Connection of logic pod . . . . . II- 4
Connection of microprocessor pod . . . . . II- 5
Connection of ML4100 to system under test II
Connection of ROM Emulator Module . . . . . II- 5
Connection of STD Bus Pod . . . . . II- 5
Connector receptacles . . . . . I- 6
Connector receptacles . . . . . I- 7
Connector receptacles . . . . . I- 9
Connector receptacles . . . . . II- 3
Continuous Compare of data IX-14
Copying data from A Memory to B Memory . . . . . IX- 8
Copying data from A Memory to B Memory IX-22

CRT . . . . . I- 7
CRT specifications . . . . . A- 4
Cursor movement . . . . . IV- 3
Data inputs . . . . . VI- 2
Data inputs specifications . . . . . A- 2
Data Interface connector pins III-11
Data Memory Option IX
Data qualification III-11
Data qualification specifications . . . . . A- 3
Delay, trigger, setup screen . . . . . III- 6
Description of ML4100. . . . . I- 3
Disassembly display . . . . . IV- 3
Disassembly display . . . . . IV- 7
Disassembly display III-21
Disassembly display specifications . . . . . A- 3
Disassembly display with Serial Data Analysis pod . . . . . XII- 9
Disconnectible probes . . . . . II- 2
Display Format setup screen III-20
Display of data . . . . . IV- 2
Display of data III-20
Display of microprocessor disassembly III-21
Display of microprocessor disassembly III-21
Display of ML4100 operating parameters III
Display of disassembly . . . . . IV- 7
Display of recorded data . . . . . IV- 3
Display of recorded data . . . . . IV- 4
Display of recorded data . . . . . IV- 6
Display of stored data IX-21
Editing ML4100 operating parameters . . . . . III- 3
Editing stored data IX-20
Error codes, ROM Emulator Module VIII-12
Error messages . . . . . E
Error warnings, audible . . . . . I- 9
Execution times, Setup/Data Memory Option IX-22
File names for B Memory IX-10
Format of data display III-20
Format setup screen . . . . . III- 7
Front panel . . . . . I- 6
Front panel . . . . . I- 7
Front panel . . . . . III- 2
Functions of ML4100 . . . . . I- 3
Functions of ML4100 . . . . . I- 4
Functions of ML4100 . . . . . I- 7
Glitch-Capture Pod (see also 4-Channel Logic Pod) V
Grounding . . . . . II- 4
Header pin assignments on connector receptacles . . . . . II- 3
Hex-to-ASCII conversion table XII-11
Inputs . . . . . VI- 2
Inputs, data, specifications . . . . . A- 2
Intel Hex format . . . . . C- 2
Intel microprocessor pod . . . . . VIII- 6
Intel microprocessor pod . . . . . VI- 3
Interface, data (see also RS-232C) . . . . . I- 9
Jumpering for STD Bus Pod . . . . . VII- 3
Keyboard . . . . . I- 6
Keyboard . . . . . I- 7
Keypad, hexadecimal . . . . . I- 7

Logic pods . . . . . I- 4
Logic pods V
Logic pods . . . . . II- 2
Logic pods . . . . . II- 3
Logic pods . . . . . II- 4
See also 32-Channel Logic Pod and 4-Channel Logic Pod
Magnification of timing display . . . . . IV- 4
Mainframe self-test III-23
Memory IX
Memory specifications. . . . . A- 3
Messages, error . . . . . E
Microprocessor disassembly display III-21
Microprocessor pod modes III-22
Microprocessor pods . . . . . I- 5
Microprocessor pods . . . . . II- 5
Microprocessor pods . . . . . III- 7
Microprocessor pods VI
Microprocessor pods, Intel . . . . . I- 5
Microprocessor pods, Intel . . . . . VI- 3
Microprocessor pods, Intel . . . . . VIII- 6
Microprocessor pods, Motorola . . . . . I- 5
Microprocessor pods, Motorola . . . . . VI- 4
Microprocessor pods, Motorola . . . . . VIII- 6
Microprocessor pods, National Semiconductor . . . . . I- 5
Microprocessor pods, National Semiconductor . . . . . VI- 8
Microprocessor pods, Rockwell I-5
Microprocessor pods, Rockwell . . . . . VI- 6
Microprocessor pods, Zilog . . . . . I- 5
Microprocessor pods, Zilog . . . . . VI- 7
Modes of pods III-22
Motorola microprocessor pods . . . . . VI- 4
Motorola microprocessor pods . . . . . VIII- 6
National Semiconductor microprocessor pod . . . . . VI- 8
Operating parameters of ML4100 III
Operating parameters of ML4100 . . . . . III- 6
Operation of ML4100 IV
Optional accessories . . . . . I- 4
Optional accessories III-23
Panel, back . . . . . I- 8
Panel, back . . . . . I- 9
Panel, front . . . . . I- 6
Panel, front . . . . . I- 7
Panel, front . . . . . III- 2
Parameters, operating, of ML4100 III
Parameters, operating, of ML4100 . . . . . III- 6
Philips PM8811A . . . . . XII- 3
Pod, 32-Channel Logic . . . . . I- 4
Pod, 32-Channel Logic . . . . . II- 2
Pod, 32-Channel Logic . . . . . II- 3
Pod, 32-Channel Logic . . . . . II- 4
Pod, 32-Channel Logic III-22
Pod, 4-Channel Logic . . . . . I- 4
Pod, 4-Channel Logic V
Pod, 4-Channel Logic . . . . . II- 3
Pod, 4-Channel Logic . . . . . II- 4
Pod, 4-Channel Logic III-22

Pod modes III-22
Pod, ROM Emulator Module . . . . . VIII- 3
Pod, Serial Data Analysis XII
Pod, STD Bus VII
Pods, microprocessor VI
Power switch . . . . . I- 9
Powerup setup storage . . . . . IX- 7
Predefined trigger sequences III-14
Price list, ML4100B . . . . . D
Printing disassembly display . . . . . IV- 7
Printing displayed data . . . . . IV- 2
Printing displayed data IX-17
Printing displayed data IX-21
Printing displayed data . . . . . VIII- 4
Printing recorded data . . . . . IV- 6
Probes . . . . . V- 2
Probes . . . . . II- 2
Probes . . . . . II- 4
Protection of setup files . . . . . IX- 7
Protection of B Memory files IX-11
Record depth III-10
Record width III-10
Recording data . . . . . IV- 2
Reference line . . . . . IV- 4
Refresh cycles for STD Bus Pod. . . . . VII- 3
Reverse assembly
See Disassembly
Rockwell microprocessor pods . . . . . VI- 6
ROM Emulator Module . . . . . I- 5
ROM Emulator Module . . . . . I- 9
ROM Emulator Module . . . . . II- 5
ROM Emulator Module III-23
ROM Emulator Module VIII
ROM Emulator Module specifications . . . . . A- 4
RS-232C Communication Control Card . . . . . I- 5
RS-232C Communication Control Card III-23
RS-232C Communication Control Card . . . . . VIII- 4
RS-232C Communication Control Card VIII-11
RS-232C serial interface . . . . . XII- 3
Safety warnings . . . . . V- 2
Safety warnings . . . . . II- 4
Saving (storing) recorded data IX
Scrolling . . . . . IV- 3
Scrolling specifications . . . . . A- 4
Searching stored data IX-18
Searching stored data IX-23
Self-test of ML4100 mainframe III-23
Sequence, trigger, setup screen . . . . . III- 6
Serial Data Analysis pod XII
Setup file directory . . . . . IX- 4
Setup Memory Option IX
Setup of ML4100 for operation III
Setup Storage screen . . . . . IX- 4
Setup/Data Memory Option . . . . . I- 5
Setup/Data Memory Option IX
Setup/Data Memory Option III-23
Single Compare of data IX-12

Special Functions . . . . . VIII- 3
Special Functions setup screen III-23
Specifications, 4-Channel Logic Pod . . . . . V- 2
Specifications, ML4100 . . . . . A
State data display . . . . . IV- 3
State data display . . . . . IV- 6
State data display III-21
State data display with Serial Data Analysis pod . . . . . XII- 8
State display specifications . . . . . A- 3
Status code prompts for microprocessor pods III-13
Status codes for microprocessor pods VI
Status setup screen . . . . . III- 4
STD Bus Pod . . . . . I- 4
STD Bus Pod . . . . . II- 5
STD Bus Pod VII
STD Bus Pod modes III-22
Storage of data in B Memory . . . . . IX- 9
Storage of recorded data IX
Switch settings for microprocessor pods VI
Text display with Serial Data Analysis pod XII-10
Threshold voltage . . . . . V- 2
Threshold voltage . . . . . II- 4
Timing data display . . . . . IV- 3
Timing data display . . . . . IV- 4
Timing data display III-20
Timing data display with Serial Data Analysis pod . . . . . XII- 7
Timing display specifications . . . . . A- 3
Trigger Delay setup screen . . . . . III- 6
Trigger Delay setup screen III-18
Trigger Sequence setup screen . . . . . III- 6
Trigger Sequence setup screen III-14
Trigger sequences III-16
Trigger setup screen III-12
Trigger words III-13
Trigger (words) setup screen . . . . . III- 6
Triggering specifications . . . . . A- 2
Triggering with Serial Data Analysis pod . . . . . XII- 4
TTL inputs . . . . . VI- 2
User-defined trigger sequences III-14
User-defined trigger sequences III-16
Voltage, threshold . . . . . V- 2
Voltage, threshold . . . . . II- 4
Warnings . . . . . I- 7
Warnings . . . . . I- 9
Warnings . . . . . V- 2
Warnings . . . . . II- 3
Warnings . . . . . II- 4
Warnings . . . . . II- 5
Zilog microprocessor pod . . . . . VI- 7

(blank)

MP MASTER
MA24071